IBM®

Federal Systems Division, Space Systems Center, Huntsville, Alabama

Laser Aiming Simulation (LASIM) Final Report

Volume II. Programmers and Users' Manual
20 May 1968

GPO PRICE $ ________

CSFTI PRICE(S) $ ________

Hard copy (HC) 3.00

Microfiche (MF) .65

ff 653 July 65

IBM No. 68-K10-0006
MSFC No. MA-004-1

N68-36227 (ACCESSION NUMBER)	(THRU)
183 (PAGES)	1 (CODE)
CR 61976 (NASA CR OR TMX OR AD NUMBER)	16 (CATEGORY)

FACILITY FORM 602

IBM

Federal Systems Division, Space Systems Center, Huntsville, Alabama

Prepared for the

GEORGE C. MARSHALL

SPACE FLIGHT CENTER

Huntsville, Alabama

Laser Aiming Simulation (LASIM) Final Report

Volume II Programmers and Users' Manual

20 May 1968

Contract No. NAS 8-21033

IBM No. 68-K10-0006

MSFC No. MA-004-1

Classification and Content Approval ______________________

Data Manager Approval ______________________

Program Office Approval ______________________

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

LIST OF ILLUSTRATIONS

NOTE: Subroutine flow charts for the LASIM Program do not carry figure numbers. The flow charts may be located by referring to the subroutine name in the Table of Contents under Paragraph 2.3.

LIST OF TABLES

SECTION 1

LASIM PROGRAM DESCRIPTION

This document is intended to supplement the Laser Aiming Simulation (LASIM) Final Report, Volume I, Mathematical Formulation, IBM Report No. 68-K10-006; and will describe the actual LASIM program, constructed from the math models discussed in Volume I. This report will describe the program functions and logic, and will illustrate the use of the LASIM program.

1.1 GENERAL DESCRIPTION

The LASIM programs are digital simulations of the pointing and tracking operation of a spaceborne Laser Communications System Experiment (LCSE). Simulation of the tracking system operation is contained within the "main program" which will be referred to throughout this report as the "LASIM program." Simulation of the pointing operation is contained within a separate program called the Pointing Control program which is discussed separately in Section 6. The Pointing Control program uses, as input, a tape generated in the LASIM (or main) program. Because of the separation of the pointing and tracking functions in the LCSE hardware operation, and also in the interest of computer running time, and because of computer memory capacity considerations, the simulation of the pointing functions is made a separate program.

The LASIM program is written in FORTRAN IV, Version 13. The machine configuration on which the LASIM program runs consists of an IBM 7094 computer, having 32K word memory, eleven tape drives, a 1403 printer and 1402 card reader. System subroutines which are used by the LASIM program are enumerated in Paragraphs 2.1 and 4.2. Tape units are referred to symbolically which provides flexibility and necessitates deleting unused files to make sufficient buffer pool storage available.

The LASIM program is modularized, as discussed in detail in Section 2, and is written almost entirely in double precision. The necessity for double precision throughout is occasioned by the extreme accuracy required in the computations.

1.2 PROGRAM FUNCTIONAL FLOW

The functions performed by the LASIM program and major program logic are shown in Figure 1-1. The operation of three major hardware systems is simulated by the LASIM program. These are the fine tracking system, the telescope control system and the spacecraft attitude control system. In addition, simulation of the physical dynamics of the mass elements comprising the LCSE (e.g. spacecraft, telescope, etc.) in response to hardware actuation is simulated.

FOLDOUT FRAME 1

FOLDOUT FRAME 2

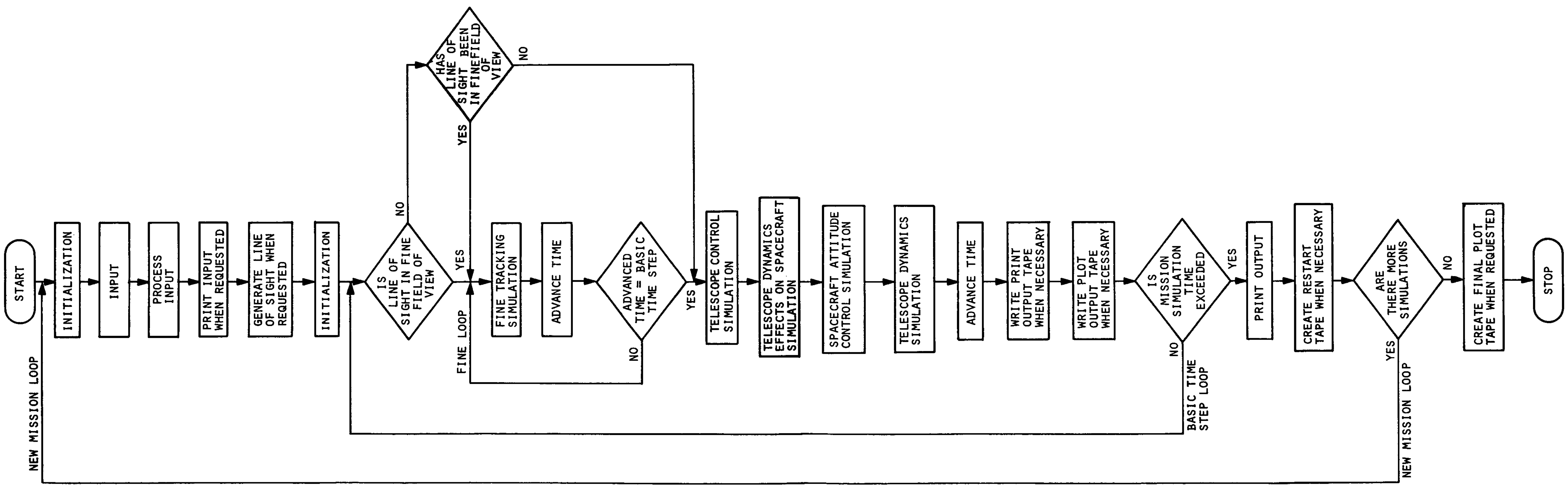


FIGURE 1-1. LASIM PROGRAM FUNCTIONAL FLOW

Wherever possible, separate subroutines have been used to simulate distinct functions within the LASIM program. Complete discussion of subroutine organization is presented in Section 2.

As shown in Figure 1-1, simulation of translational dynamics is performed to generate the orbital trajectory in which the LCSE will fly and determine the geometric relationship between an earth based laser tracking beacon and the spaceborne laser telescope. This is indicated as the line-of-sight computation. Based upon this geometrical relationship, or the accuracy with the spaceborne telescope is aligned with the beacon, control system hardware actuation is simulated. From the control system hardware operation, forces and torques are generated which act on the mass elements of the system. The effects of these control forces and torques on the mass elements are simulated through solution of the rotational dynamics equations, and act to align the spaceborne system with the ground beacon direction.

Simulation of the functions enumerated takes place in the LASIM program in the following manner. (Reference Figure 1-1) The program has three basic loops through which it cycles. These are:

- New Mission Loop
- Basic Time Step Loop
- Fine Tracking Loop.

All program functions except final plot tape creation occur in the "new mission loop." This loop begins with initialization of all variables and constants to be used in the program for a particular run or mission simulation. This loop is cycled through once for each new simulation run to be made. During a given simulation run, the hardware and dynamics equations are solved in the "basic time step loop." One pass through the "basic time step loop" increments time by one "basic time step." As programmed, this basic time step is 0.01 seconds; however, the user may select this value.

Hardware operation and program logic are dependent upon whether the ground beacon is within the fine or coarse field-of-view of the spaceborne telescope. Prior to the appearance of the beacon in the fine field-of-view, the fine tracking system is not active and its operation is not simulated. During this time the following is accomplished with each pass through the "basic time step loop."

- Coarse Optics and Telescope Control System Simulated
- Telescope Dynamics Simulated
- Spacecraft Attitude Control System and Dynamics Simulated.

- o Relative Position of Telescope with Respect to Ground Beacon Determined
- o Print and Plot Output Recorded.

Assuming normal operation of the hardware, the telescope and spacecraft will be maneuvered so as to cause the ground beacon to be acquired within the fine field-of-view. This takes place when the longitudinal axis of the telescope is aligned to within $\pm$ 1 arc minute of the ground beacon direction. With this occurrence, the fine tracking system is activated and its operation, simulated in the LASIM program. Because of the fast dynamic response of the fine tracking system, relative to the other control systems, it is necessary, in the program, to solve the system equations at a more rapid rate than afforded through the "basic time step loop." For this reason, the "fine tracking loop" was conceived and is implemented. The fine tracking system equations are solved in the "fine tracking loop" as seen on Figure 1-1. Solution of these equations is cycled five times for every pass through the "basic time step loop" of the program, once the fine field-of-view has been entered. (This rate may be changed by the user.) Should, because of some external disturbance, the beacon be lost from the fine field-of-view after it has once been acquired, the fine tracking system equations will continue to be solved in the "fine tracking loop." In this event, the fact that no light is being received by the "fine sensors" will be taken into account. However, the operation of the fine system must be updated with no light received, so that if the beacon is reacquired, the fine tracking system will react realistically.

When simulating the fine tracking system, it is necessary to update the relative position of the ground beacon with respect to the telescope at a rate faster than is done in the "basic time step loop" where this computation is normally made. So, included in each pass through the "fine tracking loop" is an update of the relative position of the ground beacon, which is used for the fine tracking system simulation.

The time increment for the basic time step loop is a multiple of the fine tracking time increment. Paragraph 3 defines program timing in more detail. As seen in Figure 1-1, the fine tracking loop is entered at the beginning of each basic time step loop. (Assuming conditions are such that the fine tracking system will be simulated.) When elapsed time in the fine tracking loop equals the basic time increment, exit is made from the fine tracking loop and the program continues.

Exit from the basic time step loop is made when elapsed time is equal to the specified mission simulation time. If it is desired to simulate another mission within the same job, the program will next recycle through the new mission loop. When all the desired mission simulations are completed and exit made from the new mission loop, the final plot tape is created from data previously stored at the end of each basic time step loop, and the program terminated.

1.3 PROGRAM FEATURES AND CONSTRAINTS

The LASIM program has been constructed to perform the dynamic simulation of the LCSE with certain features available to the user and with certain inherent constraints. The following paragraphs discuss these in general terms. Details of the options are provided in Section 4.

1.3.1 Program Features

LINE-OF-SIGHT INPUT SELECTION - The basic input to the LASIM simulation is the vector representing the line-of-sight between a ground station and the orbiting vehicle. The user may select the line-of-sight vector to be a constant, or to be calculated from solution of the dynamic equations representing the actual orbital flight of the spacecraft in relation to the ground station. Running time of the program is decreased for the constant input; however, the capability to obtain the calculated input values will be desirable for certain cases and is provided.

BATCHED JOBS - A batched job is defined as more than one simulation run, performed during one computer run. This program provides for the execution of a batched job consisting of up to ten simulation runs.

INPUT - The LASIM program provides to the user the capability of supplying input of initial values for certain critical constants and variables, exercise program options, and control program logic.

The user may supply either or both of two classes of input. These are control input and data input. Control input is used for activating options contained in the program and for controlling the logic flow. Data input may be used for initializing constants and variables. Any data words not supplied by the user will assume nominal values.

OUTPUT - The LASIM program will produce both printed and plotted output. The printed output falls into several categories as follows:

- The first output produced by the program is the diagnostic error message for each user supplied input card. This output may contain a prelist of all user supplied input cards at the user's option.

- If the trajectory generating subroutine is executed, the orbital parameters, line-of-sight vectors, and line-of-sight velocity vectors are output to tape and printed.

- A print of the values of important constants as they exist at the conclusion of a simulation run is given once for each run.

- Certain important variables will be automatically printed. Additional program variables may be printed at the user's selection.

If not otherwise specified, the program print output will be stored on logical unit 6 tape drive.

Plotted output is produced at the request of the user. Selection may be made from a set of variables, those pairs for which plots are to be created. Coordinate axis scaling will be linear unless log scaling is requested.

To make possible the restart of a simulation, data may be sorted on magnetic tape before a simulation run is terminated. The values of variables and the status of flags and switches at the time of termination are recorded on the restart tape.

A data tape containing the line-of-sight vector, line-of-sight rate vector, transfer lens positions, and the telescope-to-inertial transformation matrix will be produced for use by the Pointing Control porgram. Generation of this tape is optional.

For most simulation missions, the user will not require output for each pass through the simulation. Therefore, the program provides for a variable output frequency. Those variables which are to be printed during the simulation may be printed at any frequency specified by the user. If the print frequency is not specified, variables are printed every fiftieth pass through the basic time step loop.

Those variables which are to be plotted for a simulation run may be plotted at any frequency which does not cause more than two hundred fifty plot points to be generated by a single graph. If the simulation is to be run for n seconds, the elapsed simulation time between selection of plot points may be greater than or equal to n/250 seconds. If the plot frequency is not specified, the elapsed simulation time between selection of plot points will be set equal to n/250 seconds.

RESTART CAPABILITY - The simulation may be terminated and restarted at any time during its execution. This serves to allow the user to interrupt a lengthy simulation so that intermediate results may be validated after which the simulation may be completed. In restarted simulations, data stored on magnetic tape during a previous run is read in. The values of variables and the status of flags and switches at the time the previous simulation was terminated are thereby given to the restart simulation. The restart tape is read before the user's data is stored. Therefore, those values which the user inputs override the corresponding restart values.

TIMING - The program allows the user to control the length of time (mission simulation time, not computer usage time) for which a simulation run is to take place or mission time. Also the user may specify the basic time step increment and the fine time step increment. There are some restrictions on these specifications, however. Mission time may not be greater than 5 minutes without program modifications; and the basic time step must be an integral multiple of the fine time step.

SUBPROGRAM INHIBIT - In order that the individual hardware systems may be exercised separately, the Fine Tracking, Telescope Control, Spacecraft Attitude Control, and Control Subprograms may be inhibited at the user's option. Inhibiting a subprogram prevents execution of any of its subroutines. Use of this feature to exercise single hardware system simulations must be accomplished by some program addition to provide equivalent inputs to certain of the subprograms.

The Orbit Generator subprogram remains inhibited unless specifically requested by the user.

NOMINAL INITIALIZATION - Nominal or default values are provided for all program variables and constants. Thus, it is necessary only to specify items for which non-nominal values are desired in simulation. The Program Dictionary in Paragraph 5 lists the nominal value used for all quantities.

1.3.2 Program Constraints

Certain constraints are imposed upon use of the LASIM program due to the manner in which the program logic, scaling, etc. have been structured. The following enumerates these constraints.

- Maximum mission time which may be selected is 30 minutes or 500 seconds.
- Basic time step must be an integral multiple of the fine tracking time step.
- No more than ten jobs may be batched.
- Alternate input and output devices are restricted to logical units 12, 13, 14 and 15.
- No more than 250 points will be plotted and requests for more than twenty graphs in one simulation will be ignored.
- Requests for print of more than fifty variables and constants will be ignored.
- Print spacing and lines per page requests must be consistent.

SECTION 2

PROGRAM ORGANIZATION

This section will illustrate the manner in which the computations and logical decisions have been organized in the LASIM program to perform the functions discussed in Section 1. Whenever possible within the program, distinct functions or operations have been placed in separate subroutines. These subroutines are grouped into subprograms for description purposes. The following paragraphs will describe the functional interconnection of the various subroutines and subprograms and discuss the detailed structure of the individual subroutines.

2.1 SUBPROGRAM DEFINITION

The following paragraphs serve to define the functions performed by each subprogram in LASIM. Included are the definitions and functions of the subroutines comprising each subprogram.

2.1.1 Executive Subprogram

The Executive Subprogram is a collection of subroutines which provide data initialization, receive and process input, control simulation execution, and process output. The EXEC subroutine serves as the controlling logic for the LASIM program and the means by which input is received and output is controlled. The EXEC subroutine has as its functions:

- o Reading input cards
- o Transfer of input data to magnetic tape
- o Subprogram initiation
- o Output frequency control
- o Job termination

This routine makes possible the execution of batched jobs. It processes externally generated trajectory data, thereby providing an actual line-of-sight vector at the simulation's required time step; and for each time step of the simulation, EXEC calculates the updated components of the line-of-sight vector in the telescope frame.

The following enumerates other subroutines which, together with EXEC, comprise the Executive subprogram.

INIT1 - Subroutine INIT1 initializes all program variables and constants which may be changed in the course of execution and which therefore must be initialized to make possible the processing of multiple jobs. Subroutine INIT1 has as its functions:

- Initialization of variables and constants which are not dependent on input.
- Transfer of data from magnetic tape to representative core location.
- Calculation of variables and constants which are dependent on user supplied input.

Nominal values are supplied to all words initialized in this routine, many of which may be overridden by user supplied input. The transfer of data values from tape to core is accomplished by use of the FORTRAN Namelist function.

BLOCK DATA - Subroutine BLOCK DATA is executed at compilation time and has as its purpose the initialization of constants in common whose values are not changed in the course of a simulation. Its function is:

- Provide initialization at compilation time of constants in common using data statements.

CHKCRD - Subroutine CHKCRD tests each input card to determine if it contains input values for variables or constants, or data supplied to control the functioning of the LASIM program. Its functions are to:

- Verify that input type is valid.
- Determine input type.

While CHKCRD classifies an input card, it does not check the contents of the card for correct formating, validity, or reasonable magnitude. If the card cannot be classified, it will be ignored and job termination will occur after all input has been processed.

PROCON - Subroutine PROCON serves as the means by which information supplied by the user for controlling the functioning of the LASIM program is processed. In this subroutine control words are set with values supplied by the user. This subroutine has as its functions:

- Validation of control information
- Processing control information

When invalid control information is found, this subroutine selects the error message which explains the user's mistake. Default values are supplied for those control words which have been incorrectly specified.

PRODAT - Subroutine PRODAT transfers user supplied data from cards to tape. Its function is to:

- o Create a data tape which may be read with the Namelist function of FORTRAN.

PRODAT permits the user to supply input data cards in a free format. It does not transfer logic control information to tape, but only those values which go to programmed variables and constants.

PRIN - Subroutine PRIN provides a listing of the user's input cards. Its functions are to:

- o Sequence and print the user's input
- o Print diagnostic error messages for invalid input.

The input card images are printed only at the user's request, however, the program generated card sequence number and error messages are printed any time an error is found.

TANDR - Subroutine TANDR checks user supplied input for consistency. If inconsistent values are discovered, a related error message is selected and, to avoid job termination, default values are supplied. Its functions are:

- o Test input for consistency
- o Select error message when inconsistencies are found.
- o Supply default values to replace those found inconsistent.

This subroutine checks only those inputs whose consistency is critical for successful simulation.

OUTPRT - Subroutine OUTPRT accumulates in buffers the names and values of variables or constants which the user has selected to be printed. When all selected words have been stored, the buffers contents are transferred to an intermediate output tape. Its functions are:

- o Locate name and value of word to be printed.
- o Accumulate in buffers names and data values to be printed.
- o Transfer buffer contents to magnetic tape.

OUTPLT - Subroutine OUTPLT accumulates in a buffer data values which the user has selected to be plotted. When the buffer is filled, its contents are transferred to magnetic tape. Plot labels supplied by the user are stored on tape. Its functions are to:

- Store plot variable data in a buffer.
- Transfer buffer contents to magnetic tape.
- Store plot labels on magnetic tape.

OUTPFL - Subroutine OUTPLF creates the final plot output tapes and calls the system subroutines which produce the plotted output. The plot output tapes for all simulations in a batched job are created in this routine. This subroutine's functions are:

- Acquire and format plot labels and data.
- Invoke system routines to perform actual plotting.
- Perform log plot processing.

2.1.2 Fine Tracking Subprogram

The Fine Tracking subprogram is a collection of subroutines which simulate Fine Tracking system operation. The following enumerates the subroutines comprising the Fine Tracking subprogram.

FINE - Subroutine FINE serves as the main subroutine for the Fine Tracking subprogram. Its purpose is to determine when the line-of-sight vector is in the fine field-of-view and to provide logic necessary for Fine Tracking subprogram execution or immediate return to subroutine EXEC when fine acquisition has not been made. Functions of subroutine FINE are:

- Determine if the line-of-sight vector is in the fine field of view.
- On fine acquisition initiate execution of Fine Tracking simulation.
- Compute transfer lens velocity.
- Update parameters changed during fine system execution.

CALC2 - Subroutine CALC2 simulates Fine Tracking system dynamics and control system operation. Its functions are:

- Compute transfer lens position error.
- Compute transfer lens driving voltage.
- Compute transfer lens motion.

XYCURV - Subroutine XYCURV provides the basis for the calculation of transfer lens voltage. Its function is to:

- Calculate the light energy fraction used in generating transfer lens voltage.

The image position in the f/70 plane is used to determine the corresponding light energy fraction from a curve stored in this subroutine.

RNG - Subroutine RNG generates random numbers for use in the selection of log-normally distributed light energy. Its function is to:

- o Generate random numbers in the inerval (0, 1).

The sequence of numbers created by this program has a uniform distribution; however, the sequence will be repeated each time the program is reinitialized.

CALC3 - Subroutine CALC3 generates a set of light energies with a log-normal distribution for use in the transfer lens simulation. Its functions are:

- o Calculate mean and standard deviation of the log-normal distribution for a given light energy distribution.
- o Record light energy from the log-normal distribution.

INTENS - Subroutine INTENS uses random numbers to select light energy from a log-normally distributed set. Its function is to:

- o Select and store randomly light energy from a log-normally distributed collection.

Computation of transfer lens voltage is dependent on this light energy.

2.1.3 Telescope Control Subprogram

The Telescope Control subprogram simulated Telescope Control System operation. The simulation occurs in one subroutine, TELCON.

TELCON - Subroutine TELCON simulates operation of the Telescope Control system. Its functions are:

- o Compute telescope position errors.
- o Compute telescope angular rates.
- o Generate telescope motor torques.

A coarse optics sensor model is evaluated to determine telescope position errors. To simulate the Telescope Control system hardware the subroutine solves difference equations derived using Tustin's method.

2.1.4 Control Subprogram

The Control subprogram, consisting of two subroutines, computes telescope dynamics and provides an update of the (T2B) and (T2I) transformation matrices. The dynamics computations are performed in one subroutine (CONTRL) with two entry points.

CONTRL - Subroutine CONTRL performs the required telescope dynamics computations and updates the matrix (T2B) which transforms vectors from the telescope frame to a spacecraft body frame. It has as its functions:

- o Compute experiment package inertias.
- o Compute telescope gimbal angles, gimbal angle rates, and accelerations.
- o Update (T2B) transformation matrix.

The last two functions are executed under the second entry point called CONTL2. Trapezoidal integration is used for the calculations of the telescope gimbal angles and their rates.

DIRCOS - Subroutine DIRCOS computes the rate of change of the elements of the matrix (T2I) which transforms vectors from the telescope frame to the intertial frame. The elements of the matrix are then integrated over the basic time step increment. The functions of this subroutine are:

- o Update derivatives of the (T2I) transformation matrix elements.
- o Integrate to obtain (T2I) transformation matrix elements.

The integration technique used here is standard fourth order Runge-Kutta.

2.1.5 Spacecraft Attitude Control Subprogram

The Spacecraft Attitude Control subprogram simulates the spacecraft control moment gyro (CMG) system and evaluates spacecraft attitude dynamics, taking into account external disturbance torques. This subprogram consists of two subroutines and three "function" subroutines. The following describes these.

SCATT - Subroutine SCATT simulates the CMG control system and solves the spacecraft attitude dynamics equations. It has as its functions:

- o Solve CMG control system equations.
- o Solve CMG dynamic equations.
- o Solve spacecraft attitude dynamics equations.

This subroutine uses Runge-Kutta-Gill integration techniques for the solutions of the CMG and attitude dynamics equations.

ASTROM - Subroutine ASTROM provides torques resulting from external disturbances which act on the spacecraft. Its function is:

- o Compute external disturbance torques related to astronaut motion.

The computations produce a prestored disturbance torque profile. The profile is fixed and its change requires program modification.

F1, F2, F3 - Function subroutines F1, F2 and F3 provide the capability for computing repetitively, lengthy arithmetic functions occurring in the Spacecraft Attitude Control subprogram. Their function is:

- o Compute cross coupling torques on the spacecraft due to control moment gyro motion.

2.1.6 Orbit Generating Subprogram

The Orbit Generating subprogram is a collection of subroutines which generate a spacecraft orbit and compute the line-of-sight vector from ground station to a spacecraft in the orbit. Fifty values for each of the components of the line-of-sight position and velocity vectors are computed at a constant time spacing. The LSQPF routine is called then by EXEC to fit a curve through the fifty points and furnish the coefficients for a third order polynomial in time, which is then solved at the basic time step loop rate in the LASIM program. The subprogram functions only at the user's request. The following enumerates the subroutines comprising the Orbit Generating subprogram.

ORBGEN - Subroutine ORBGEN serves as the main subroutine for the Orbit Generating subprogram. Its purpose is to provide computations and initiate logic which will result in the output of the line-of-sight vector and its velocity. The functions of subroutine ORBGEN are:

- o Compute and print orbital parameters.
- o Compute ground station position.
- o Initiate execution of orbit generation and computations for spacecraft position and velocity and line-of-sight position and velocity.
- o Store output.

PVINO - Subroutine PVINO serves as the means by which the initial spacecraft position and velocity in the standard inertial frame are determined. Its functions are:

- o Compute initial spacecraft position in previously defined orbit.
- o Compute initial spacecraft velocity.

DFLCW - Subroutine DFLCW serves to compute by integration the vehicle position and velocity over the time step specified to the Orbit Generating subprogram. Its functions are:

- o Compute new vehicle position.
- o Compute new vehicle velocity.

Cowell's technique is employed to perform the integration.

DERIV - Subroutine DERIV evaluates the orbital motion equations for the new vehicle acceleration from orbital parameters previously calculated. Its function is:

- o Compute spacecraft acceleration in previously determined orbit.

The orbital equations of motion are standard and account for the second through fourth zonal harmonics of the earth's gravitational potential under the assumption that the earth's gravitational attraction provides the only source of vehicle acceleration.

ANGLES - Subroutine ANGLES computes the line-of-sight vector and its velocity. Its functions are:

- o Compute components of the line-of-sight vector in the inertial frame.
- o Compute components of the line-of-sight velocity vector in the inertial frame.

OPUT - Subroutine OPUT records and prints output created during each time step through the Orbit Generating subprogram. Its functions are:

- o Record on magnetic tape line of sight position and velocity components.
- o Print spacecraft and ground station position and velocity.

2.1.7 Utility Subroutines

Several routines and functions may be categorized as utility subroutines. These subroutines are employed throughout the LASIM program and do not logically fit into any of the major subprograms. The following enumerates these routines.

MAMULT - Subroutine MAMULT performs the multiplication of two three-by-three matrices. Its function is to:

- o Multiply two matrices and store the product.

This routine improves on the system subroutine in that the dimensions of the matrices are known.

CROSS - Subroutine CROSS calculates the cross-product of two vectors. Its function is to:

- o Calculate the cross product of two supplied vectors and store the result.

LSQPF - Subroutine LSQPF is a system subroutine which fits a polynomial of order one through seven to a given set of points by the method of least squares. To reduce the run time of the Orbit Generating subprogram, this subroutine is invoked to provide polynomial coefficients for the line-of-sight vector component equations. Values for the line-of-sight vector components will be obtained from the polynomial as a function of time at the frequency required by the simulation. Its functions are:

- o Provide third order polynomial coefficients for each component of the line-of-sight position vector as a function of simulation time.

This subroutine fits a curve through only fifty points.

QUIK3V - This system subroutine initiates production of a plot tape for the SC-4020 Plotter. The functions of subroutine QUIK3V are:

- o Compute minimum and maximum values of the X and Y arrays.
- o Call QUIK3L, the system subroutine that produces the graphs and connects the plotted points with a straight line.

The plot tape will be A8 and should not be unloaded or changed by attaching SYSLB2 as another tape.

SMXYV - Subroutine SMXYV is a system subroutine which provides the capability for logarithmic plotting. Its function is:

- o Initiate logarithmic plot mode.

CLEAN - Subroutine CLEAN is a system subroutine which dumps the plot buffer. Its functions are:

- o Store plot data on SC-4020 plot tape.
- o Write current job sequence number in last frame.
- o Write an end of file on the SC-4020 tape.

The following arithmetic system subroutines are found throughout the LASIM program.

DCOS - Computes the double precision cosine of the angle in radians supplied as its argument.

DSIN - Computes the double precision sine of the angle in radians supplied as its argument.

DSQRT - Computes the double precision square root of the argument.

DLOG - Computes the double precision, natural logarithm of the argument.

DABS - Computes the double precision absolute value of the argument.

DSIGN - Computes the product of the absolute value of the first argument and the sign of the second.

ABS - Computes the real, absolute value of the argument.

MOD - Computes the integer remainder obtained when the first argument is divided by the second.

ACOS - Computes the single precision angle in radians whose COS is the real number supplied as its argument.

ASIN - Computes the single precision angle in radians whose SIN is the real number supplied as its argument.

ATAN - Computes the single precision angle in radians whose TANGENT is the real number supplied as its argument.

ATAN2 - Computes the single precision angle whose TANGENT is the quotient obtained by dividing the first argument by the second.

2.2 PROGRAM STRUCTURE

Figure 2-1 illustrates the grouping and interconnection of the subroutines within the LASIM program. The following will describe the execution of the program in terms of the subroutines called, as shown on Figure 2-1.

The BLOCK DATA subroutine shown on Figure 2-1 functions at compilation time to initialize constants. After execution begins, all subprograms are under control of the EXEC subroutine which is part of the Executive subprogram. Execution of a simulation run starts in subroutine EXEC. The first subroutine called by EXEC is INIT1, where variables and constants are initialized to nominal values. After this is done, control is returned to EXEC from which any user supplied input cards are read in. Subroutine CHKCRD is called next to classify the input as either control input or data input. Depending upon the classification, either PROCON or PRODAT subroutines are called to process control or data input respectively.

Subroutine PROCON alters the program logic by setting appropriate switches and flags based on the control input. Subroutine PRODAT creates a data tape which is read into the program later using the FORTRAN Namelist routine to set the appropriate variables or constants to the desired value.

Subroutine PRIN is next called by EXEC to print the input card images and any diagnostic messages indicated in either PROCON or PRODAT. Any invalid control or data word entries on input cards will cause termination of the run but not before all cards have been processed so that the reader for input errors can be completed.

The above processing is repeated for each input card until an END card is read. END is treated as a control word and PROCON sets a switch to terminate card reading and proceed with program execution.

At the end of a completed simulation run, control is transferred back to this point in EXEC as shown in Figure 2-1. A /* card, when read by EXEC, signals completion of all jobs and causes creation of a final plot output tape, after which execution is terminated.

Proceeding through the program execution shown on Figure 2-1, if a control word has been read to indicate the current job is a restart, subroutine RESTAR is called from EXEC and data contained on the restart tape brought into core via Namelist. Next, INITR is called by EXEC if any data input has been supplied by the user. If so, these data are read into the appropriate location via the FORTRAN Namelist routine.

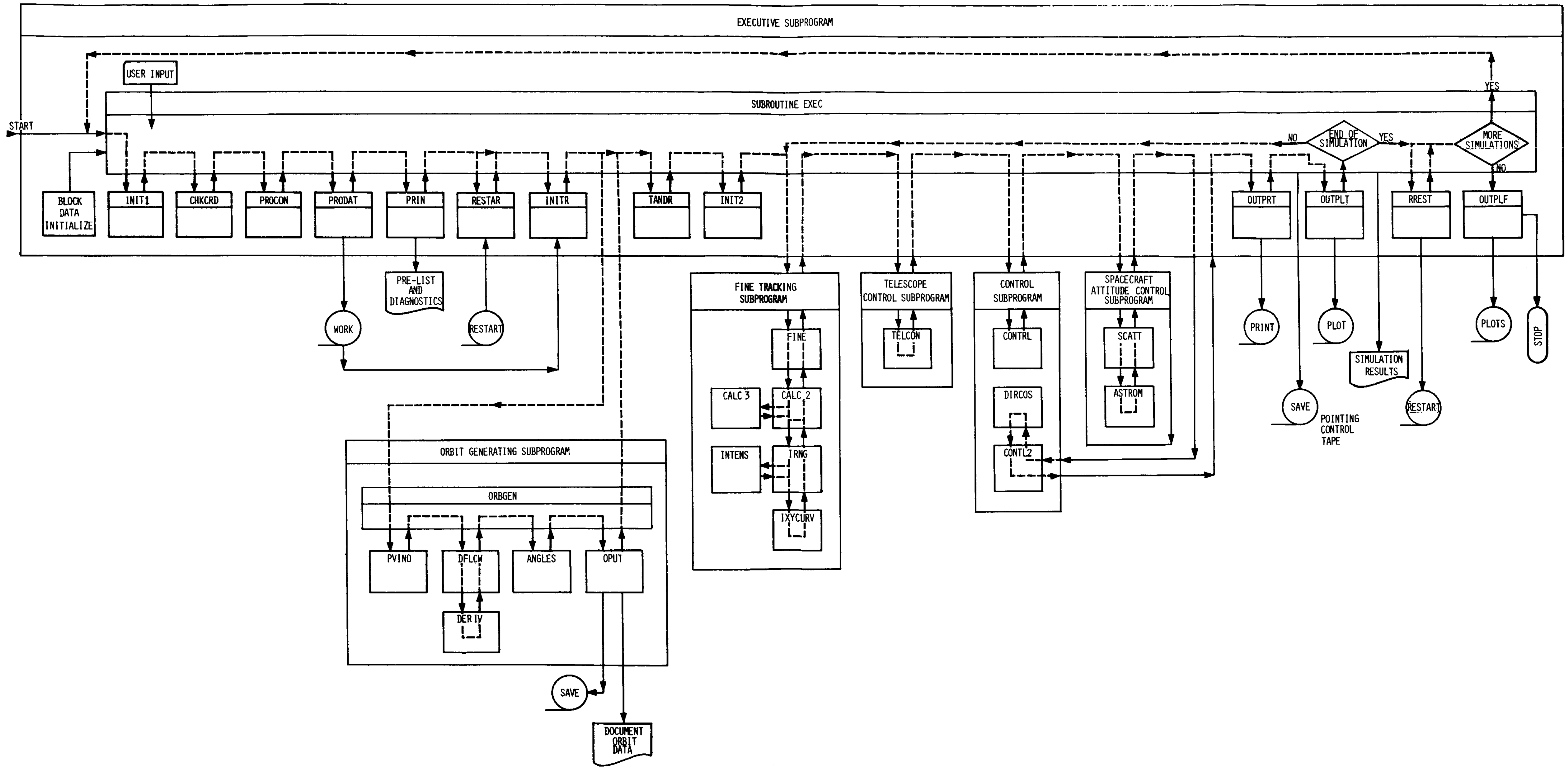


FIGURE 2-1. LASIM PROGRAM STRUCTURE

If the line-of-sight vector is to be generated as a function of the Orbital Trajectory, subroutine ORBGEN in the Orbit Generator subprogram is next called. ORBGEN serves as the calling routine for the Orbit Generator subprogram. Through its execution, the Orbit Generator produces fifty points for the line-of-sight and line-of-sight rate components. Next, a curve-fit routine LSQPF is called by EXEC to obtain the coefficients of a third order polynomial which will reproduce these points. The polynomial is then evaluated in EXEC, with elapsed time as the variable, each basic time step loop through the program, to obtain the line-of-sight input. This is done because the line-of-sight must be updated every 0.01 second in the LASIM program. Running the Orbit Generator subprogram at this small time step would result in prohibitly long running time for the simulation.

As seen on Figure 2-1 the next routine called is subroutine TANDR. In this subroutine, certain checks (described in paragraph 2.2) are made on the data input and correction made if necessary.

Next, subroutine INIT2 is called by EXEC. In INIT2, all initialization calculations which are dependent upon input data are made. Control is again returned to EXEC after execution of subroutine INIT2.

The simulation loop begins next with the calling of subroutine FINE from EXEC. FINE controls the other subroutines making up the Fine Tracking subprogram. In FINE, a test is made to determine if the ground beacon is in the fine field-of-view. If so, transfer lens motion is simulated in subroutine CALC2. Solution of the fine tracking system equations is performed five times (the number of cycles through the fine tracking loop, while nominally 5, may be changed by the user) in the Fine Tracking subprogram. If the ground beacon is not in the fine field-of-view control is returned to EXEC.

Subroutine EXEC next calls subroutine TELCON, wherein the telescope control system is simulated. In TELCON, the output of the experiment package torquers is determined for use in evaluating telescope dynamics. Subroutine CONTRL is next called to predict ahead the telescope angular velocities and compute telescope inertias which vary with gimbal angle. Also computed at this time are coupling torques acting on the spacecraft produced by the telescope dynamics.

Simulation of the spacecraft control moment gyro (CMG) control system and spacecraft attitude dynamics occurs next in the Spacecraft Attitude Control subprogram. Subroutine SCATT is called by EXEC and the CMG Control System equations and spacecraft attitude dynamics equations are solved. SCATT calls ASTROM, which provides torques resulting from external disturbances. Re-entry to the

Control subprogram follows where telescope dynamics are computed and the telescope-to-body transformation matrix updated under the second entry point, CONTL2. Subroutine DIRCOS is called to update the derivatives of the telescope-to-inertial transformation matrix elements and to integrate matrix elements over the total simulation time step. Control is then returned to subroutine EXEC where two decisions are made concerning output.

If it is time for print output, EXEC calls subroutine OUTPRT which stores on an intermediate tape, simulation results from the preceding time step which are to be printed. Then, if it is time for plot output, EXEC calls OUTPLT which stores plot output on an intermediate plot tape. EXEC will also output the necessary data on tape for the Pointing Control program, if requested. If simulation time has not been exceeded, control is passed back to the beginning of the simulation loop.

The simulation subprograms are re-executed, repeating the processes discussed above over the next time step. When simulation time expired, the print output tape created in subroutine OUTPRT is copied on the system output tape. If a restart of the simulation is to occur, a restart tape is created, containing all necessary variables and constants. If there are more simulation missions to be run, control is passed back to the start of subroutine EXEC, initialization is again performed, more cards are read in, and the total program is rerun for the new mission. After all missions have been run, the final SC-4020 plot tape is created from the OUTPLT tape and the program is terminated.

2.3 SUBROUTINE DESCRIPTION

The following paragraphs describe each subroutine used in the LASIM program. These descriptions illustrate how the subroutines are organized to accomplish the functions enumerated in Paragraph 2.1. Flowcharts of the subroutines are shown to illustrate subroutine logic and operation.

2.3.1 EXEC Subroutine

All subprograms are under control of EXEC which assumes control at the start of program execution and calls for subroutines at the appropriate time. At the start of program execution, subroutine INIT1 is called where those variables and constants whose values may have been changed by a previous simulation run including those changed by prior inputs are initialized to nominal or default values. Following this initialization, EXEC will read input cards supplied by the user.

Subroutine CHKCRD is called unless the card read is a /* card. The /* card signals the completion of all jobs and when it is read, the final plot output tape is created and execution is terminated. If the input card is classed as control, subroutine PROCON is called. If it is classed as data, subroutine PRODAT is called. Any invalid data or control words will cause termination of simulation but only after all input has been processed so that the search for input errors can be completed. If an error has been found in the input card, or if the user has requested a listing of his input, subroutine PRIN is called where diagnostics and card images are printed.

If the input card is not an END card, another card is read and processed as described above. When all input has been received as indicated by an END card, and no input errors have been found, the DATA tape is formatted and rewound.

If the current computer run is a restarted simulation, the required data saved from the previous simulation is acquired from magnetic tape. If there has been any user supplied input data, it is read from tape in INITR, the second entry point of subroutine INIT1. If the line-of-sight vector is to be derived from a described orbit, the EXEC calls ORBGEN; otherwise, constant elements are used to produce a test case line-of-sight vector.

The EXEC next calls subroutine TANDR where consistency of input is checked. Inconsistent values are corrected, and the user is informed of corrections.

The sequence in which the major hardware system subroutines are called is fixed; however the omission of any one may be accomplished through the control words and necessitates a series of

EXEC decisions to determine the activation of each simulation subprogram. An affirmative decision to simulate the Fine Tracking System results in immediate transfer of control to the FINE subroutine. After fine tracking simulation has been accomplished, program control is returned to the executive program. Entry to the TELCON subroutine follows, if it has not been deactivated, where telescope control is simulated.

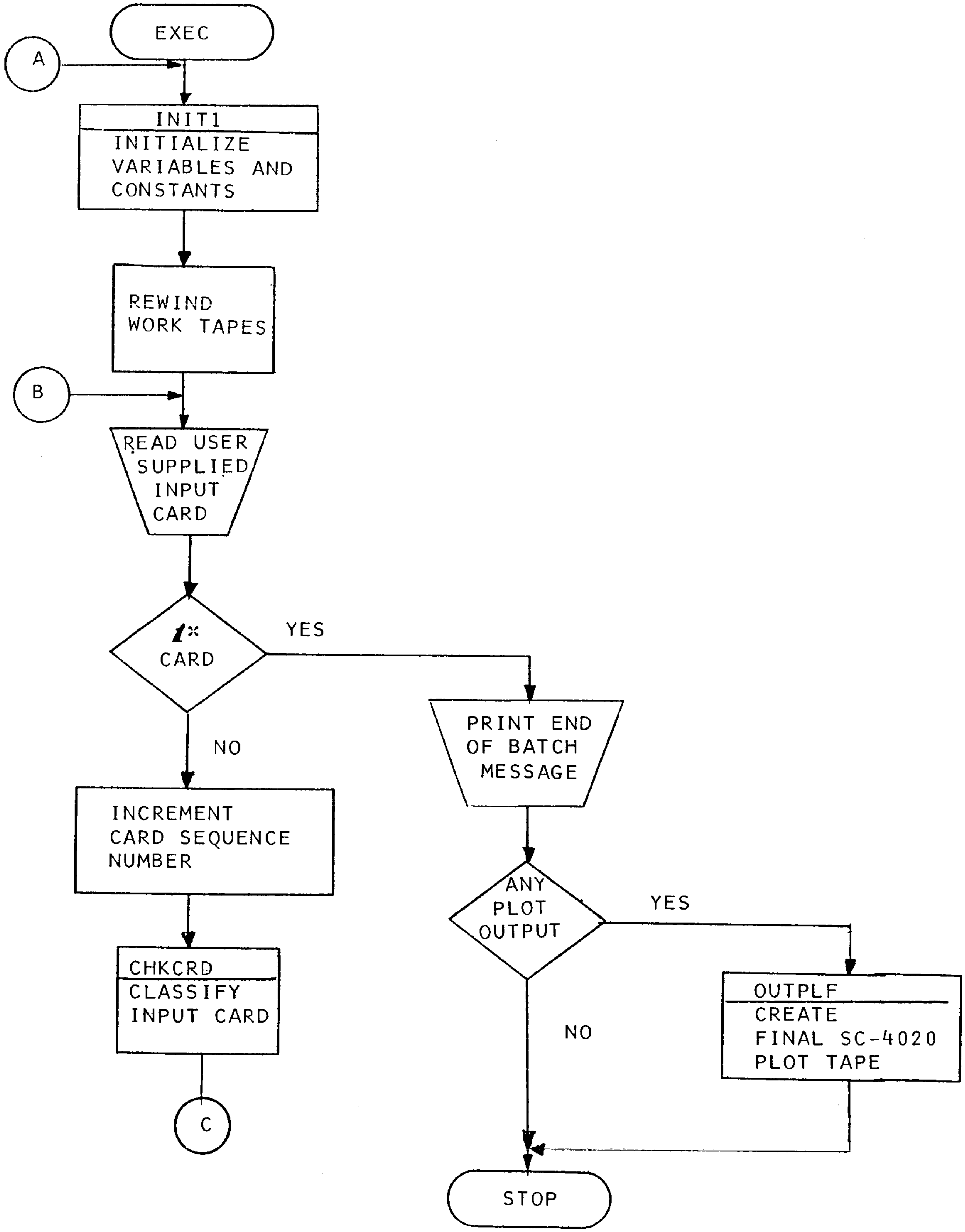
The spacecraft attitude control subprogram is entered by calling subroutine SCATT next, provided no request has been received for its omission. After execution of SCATT, entry from the EXEC is then made to the second segment of the Control subprogram to compute telescope dynamics.

If a request has been received to create a pointing control tape, the requisite information is next stored on magnatic tape.

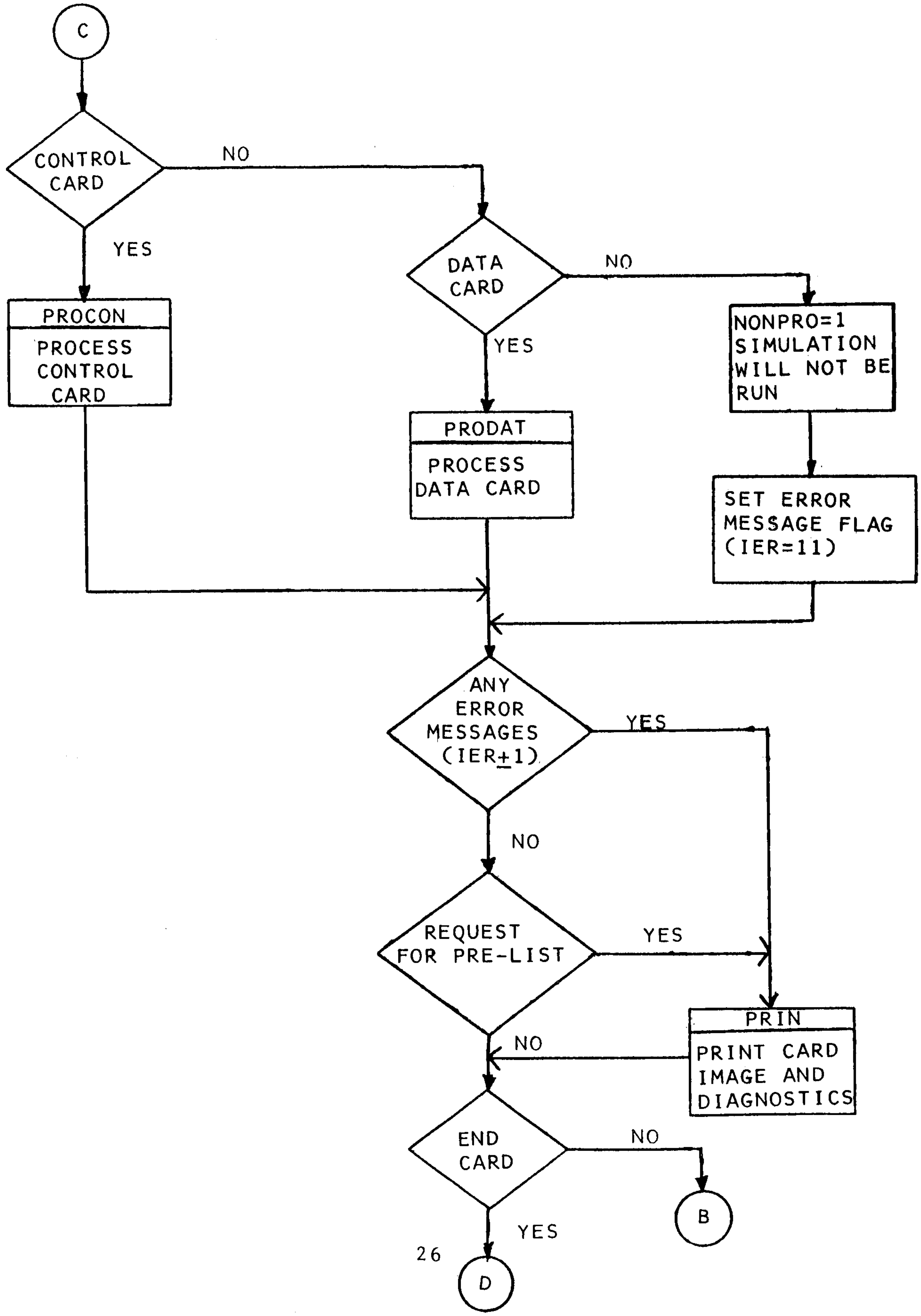
Contingent on output frequency, the results accumulated during the previous pass through the program are stored on magnetic tape. This applies to both printed and plotted output. Subroutine OUTPRT is called for print and OUTPLT for plots.

At this juncture, simulation is complete through the next incremental time step and all output data has been saved on the appropriate device. If elapsed time neither equals nor exceeds requested simulation time, control is transferred back to the first of the activated hardware subroutines, nominally FINE, for simulation through another basic time step. If simulation time has expired, the second entry point, OUTPL, of OUTPLT is entered to record any remaining plot data. If a restart of the current simulation is to be made, the necessary data is saved on tape under RREST, the second entry point of subroutine RESTAR. Subroutine EXEC then creates any required extra copies of print output. If additional simulations are to be executed, control is passed back to the first initialization segment of the subroutine. The absence of additional runs results in termination of the program.

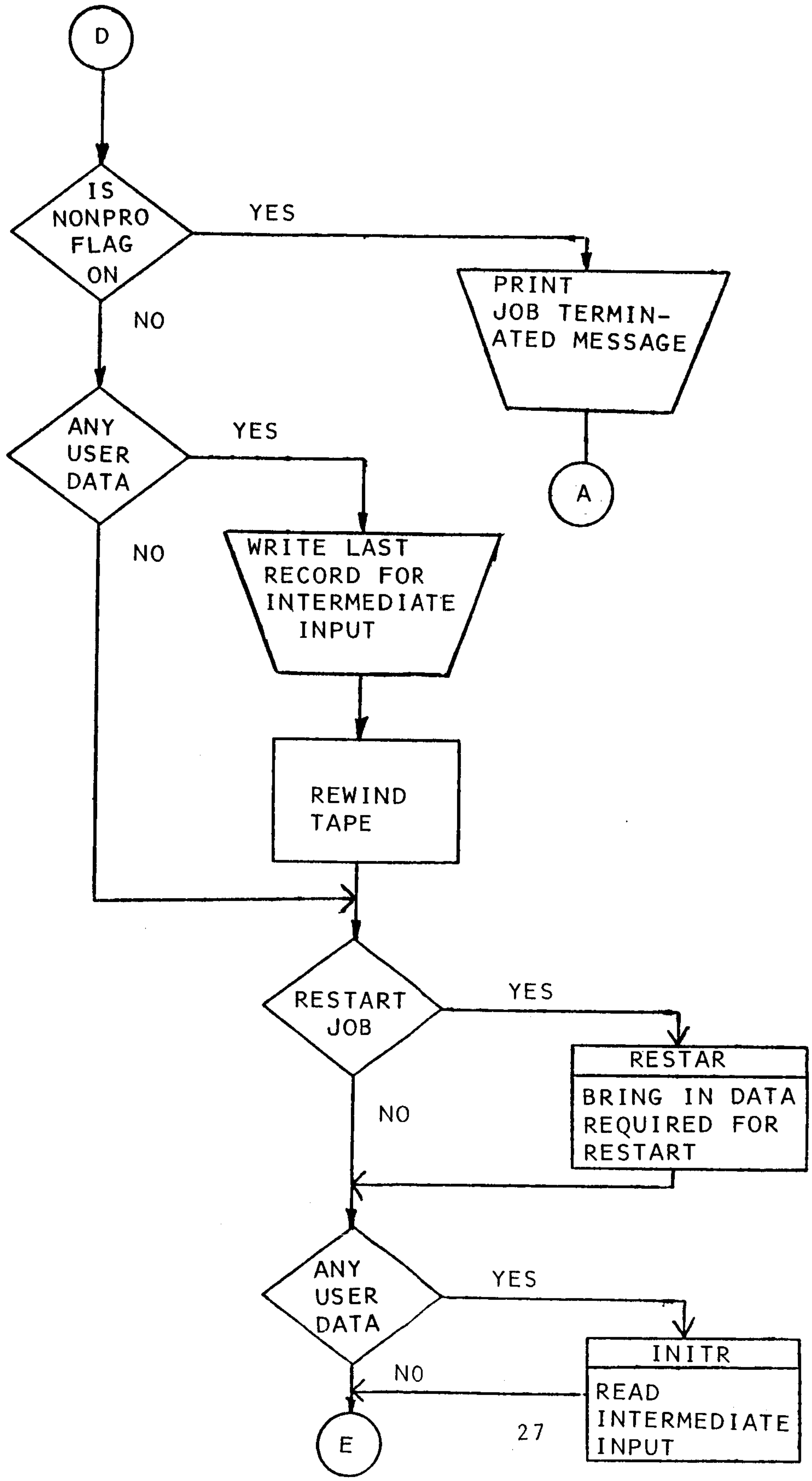
EXEC FLOWCHART

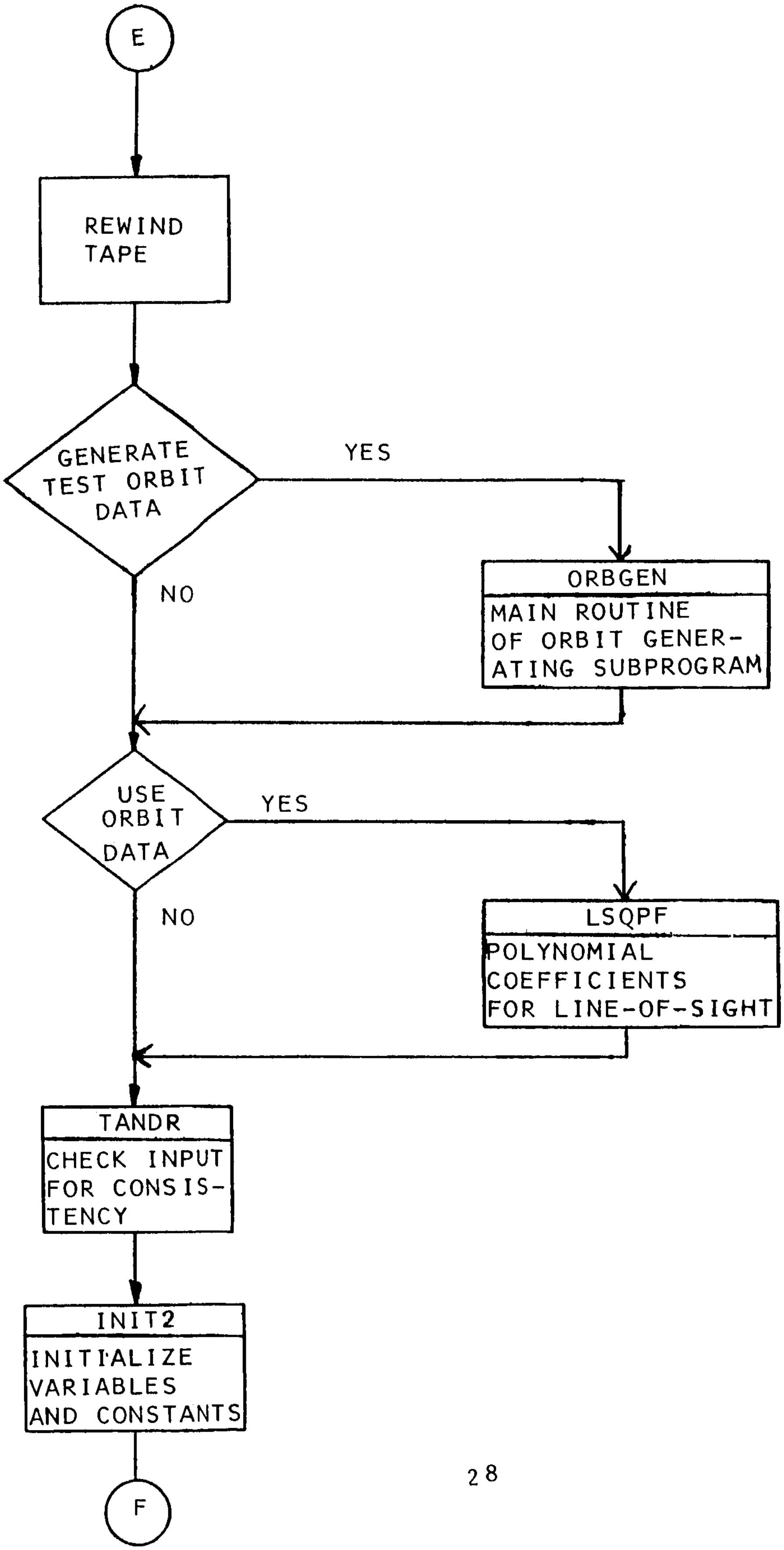


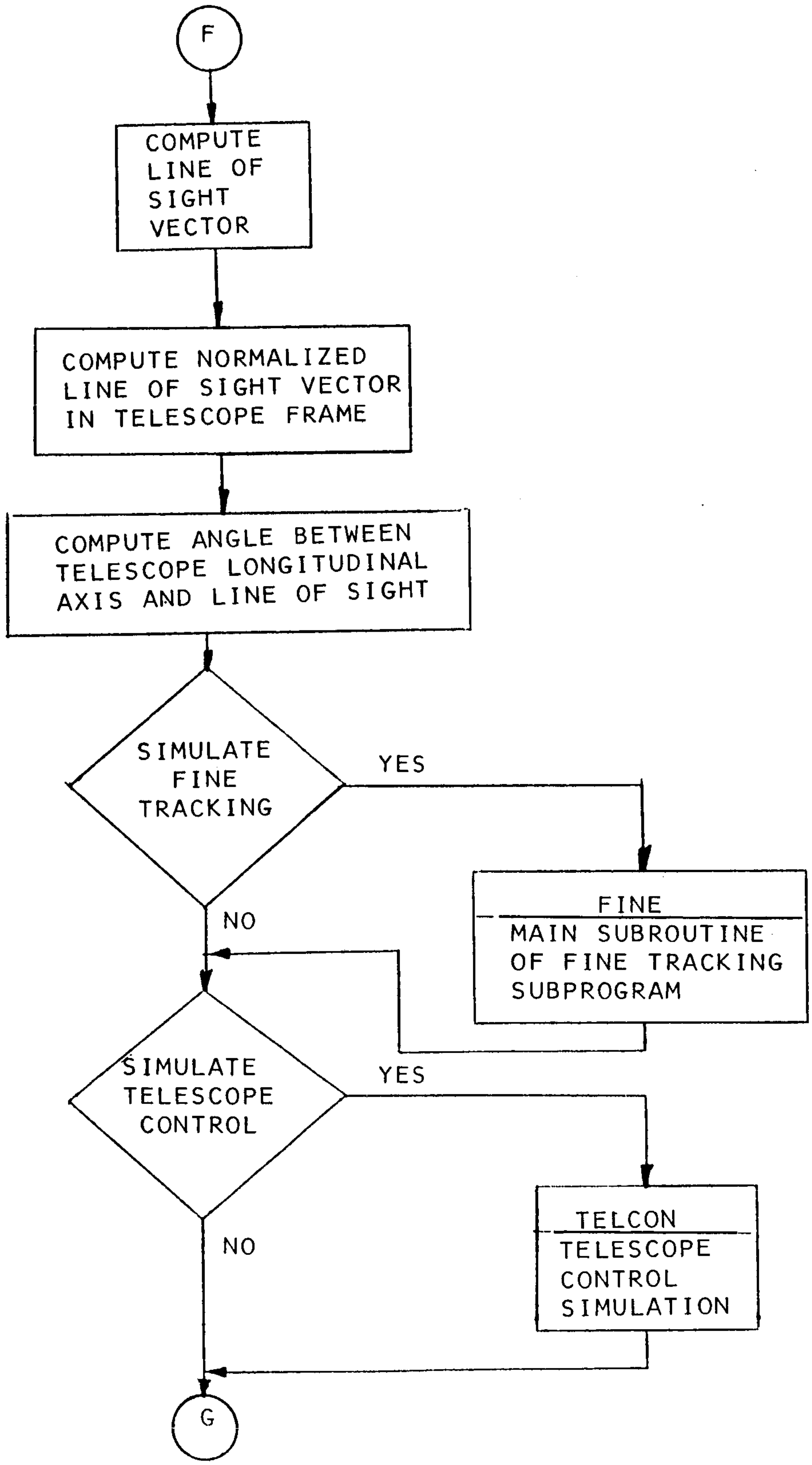
EXEC FLOWCHART (CONTINUED)



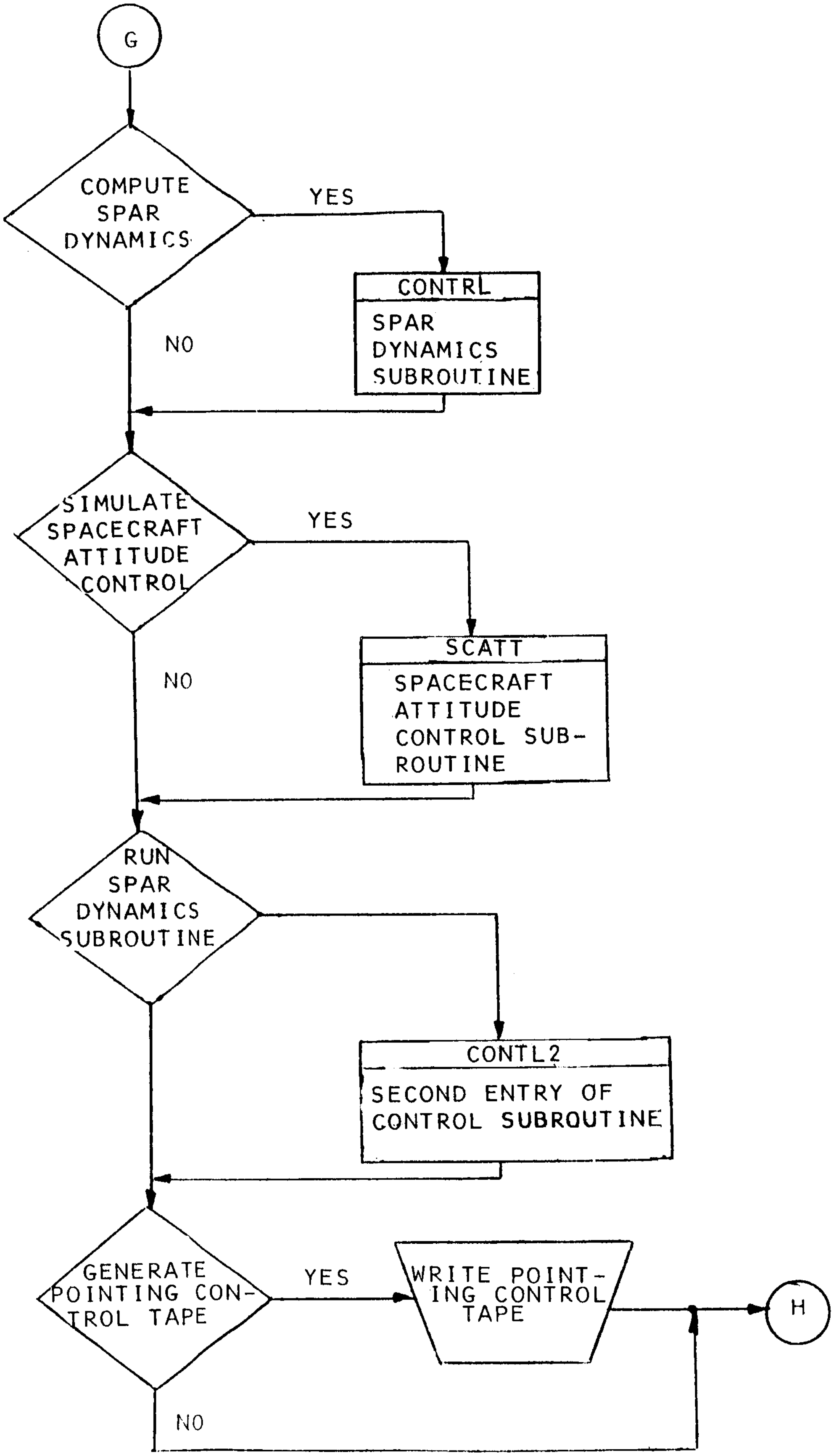
EXEC FLOWCHART (CONTINUED)



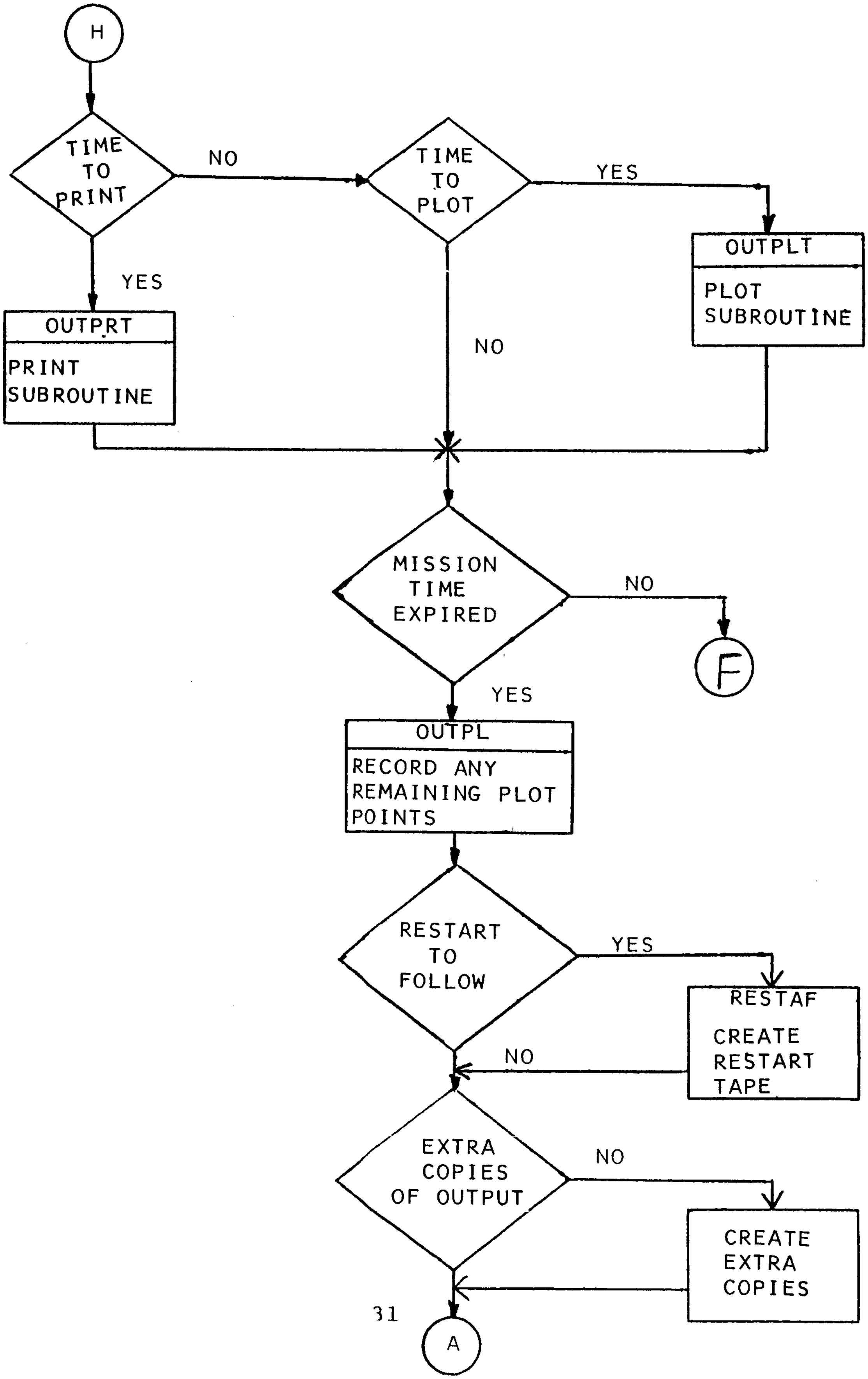

E
REWIND TAPE
GENERATE TEST ORBIT DATA
YES
NO
ORBGEN
MAIN ROUTINE OF ORBIT GENERATING SUBPROGRAM
USE ORBIT DATA
YES
NO
LSQPF
POLYNOMIAL COEFFICIENTS FOR LINE-OF-SIGHT
TANDR
CHECK INPUT FOR CONSISTENCY
INIT2
INITIALIZE VARIABLES AND CONSTANTS
F


F
COMPUTE LINE OF SIGHT VECTOR
COMPUTE NORMALIZED LINE OF SIGHT VECTOR IN TELESCOPE FRAME
COMPUTE ANGLE BETWEEN TELESCOPE LONGITUDINAL AXIS AND LINE OF SIGHT
SIMULATE FINE TRACKING
YES
FINE
MAIN SUBROUTINE OF FINE TRACKING SUBPROGRAM
NO
SIMULATE TELESCOPE CONTROL
YES
TELCON
TELESCOPE CONTROL SIMULATION
NO
G

EXEC FLOWCHART (CONTINUED)



EXEC FLOWCHART (CONTINUED)



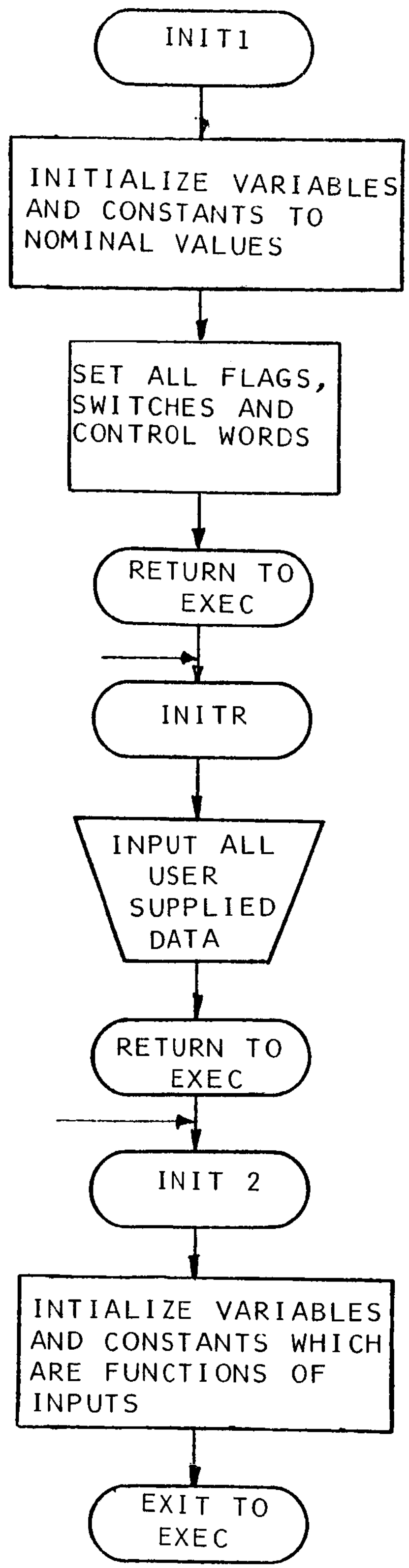
2.3.2 INIT1, INITR, INIT2 Subroutines

The initialization program consists of three sections. The first section of code under subroutine INIT1 provides initialization of variables and constants, some of which may be subsequently overridden by user input. The input of these values is optional and therefore all of those which may be changed are initialized. Failure to input any value initialized in this section will result in its remaining at the default value. Each control word is initialized to its nominal value. This section of code also initializes all program logic flags and switches. Upon completion of these initializations, return is made to EXEC.

The second section under entry point INITR serves to read in data supplied by the user from the intermediate tape created in the PRODAT subroutine. Upon completion of this section, all user supplied data has been stored in memory and return is made to EXEC.

The final section of initialization under entry point INIT2 initializes those variables and constants which are functions of variables and constants which may have been received as input. On completion of this section, all initialization is complete and execution resumes in EXEC.

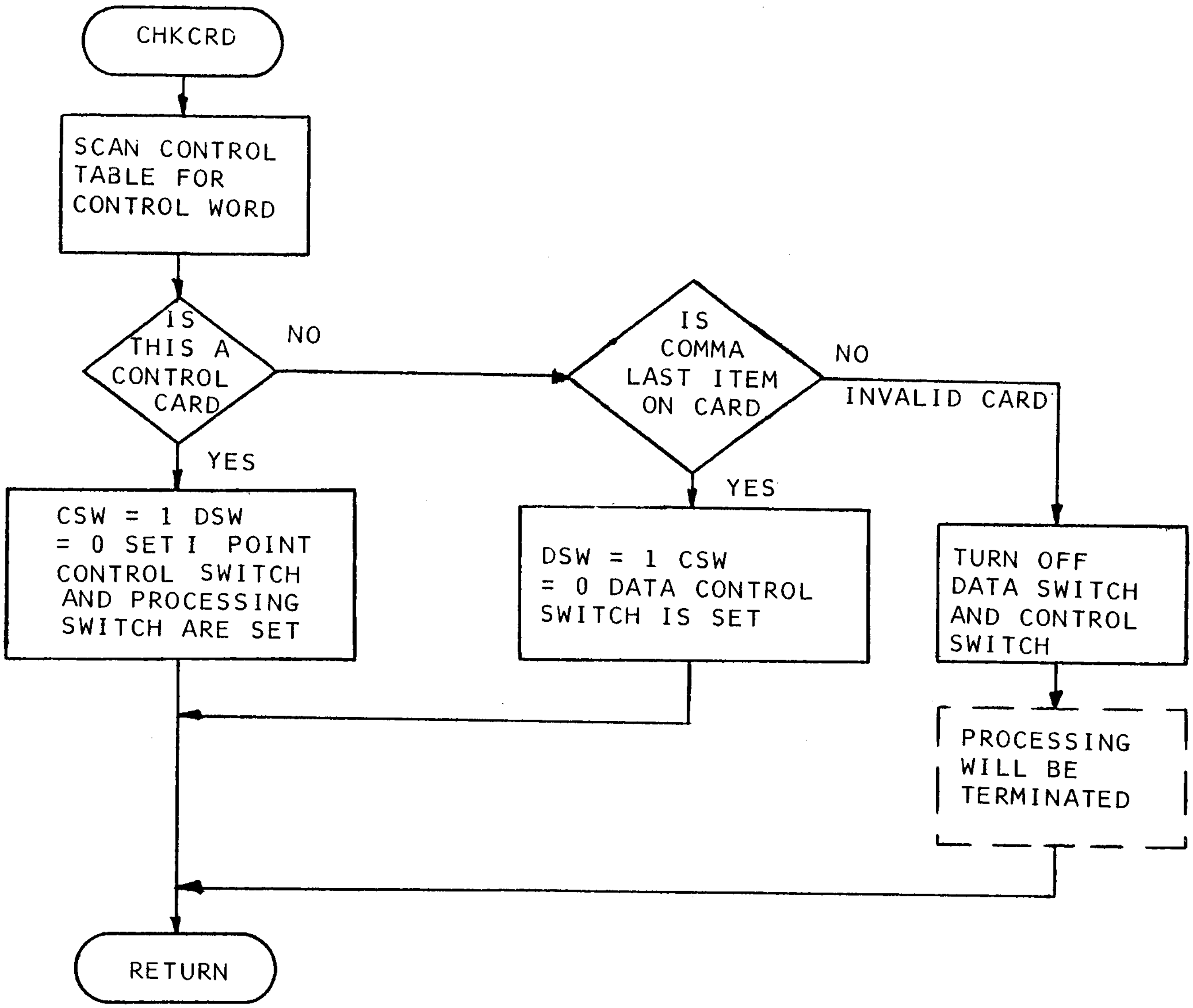
INITIALIZATION FLOWCHART



2.3.3 CHKCRD Subroutine

This subroutine is called by EXEC after each input card image has been read from the input tape. A table of valid control words is scanned to determine if the first word on the card is a valid control word. If the first word (columns 1-6) is in the control table, the control switch (CSW) is turned on, the data switch (DSW) is turned off and the processing switch (IPOINT) is set to cause the appropriate processing in subroutine PROCON. If the input card is not control information, it is scanned from column 80 down to insure that a comma follows the last entry. No other tests are made to validate the card. The absence of a comma after the last data entry results in termination of the job following all input processing. A blank card will also terminate the program. Execution of this program is followed by return to EXEC.

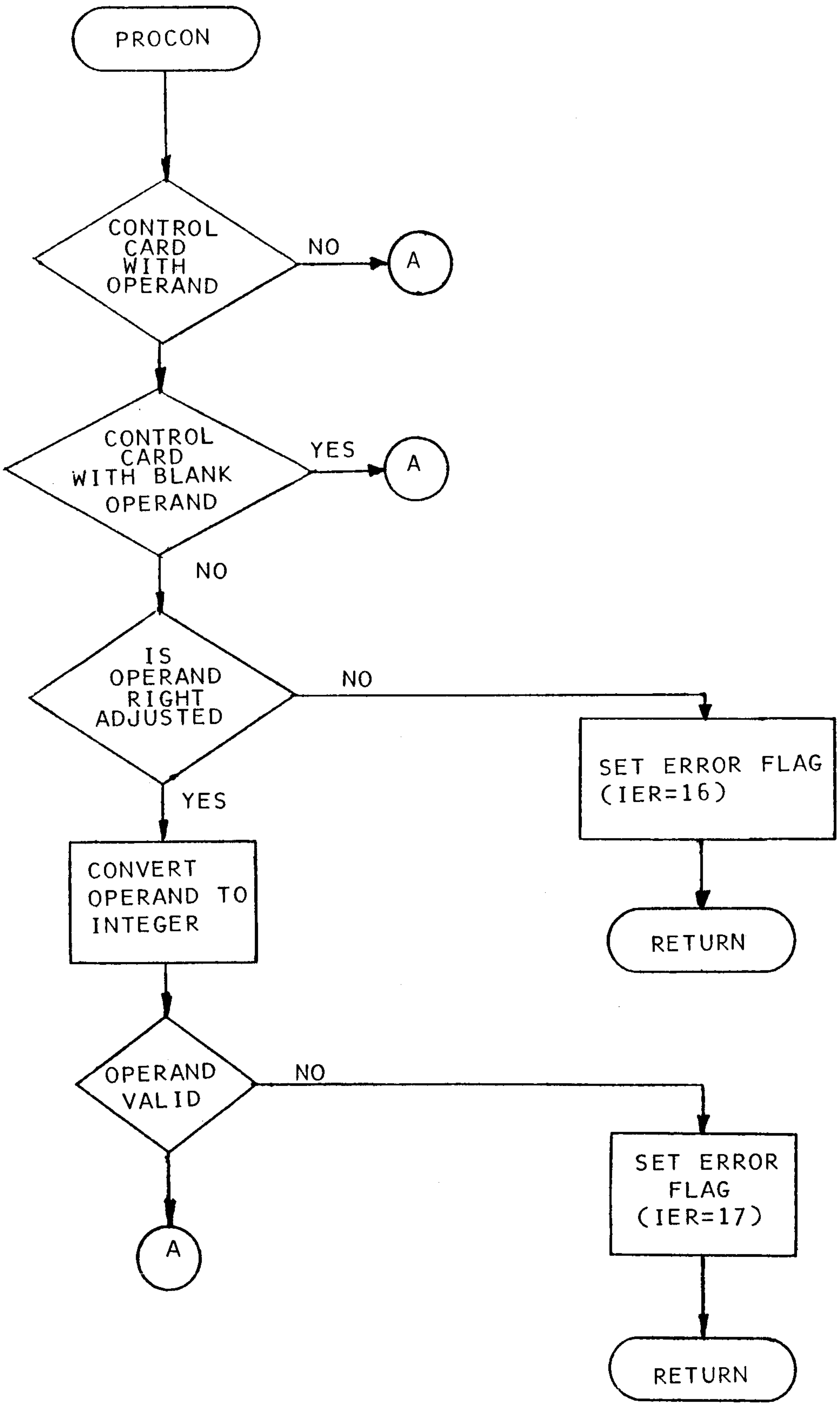
CHKCRD FLOWCHART



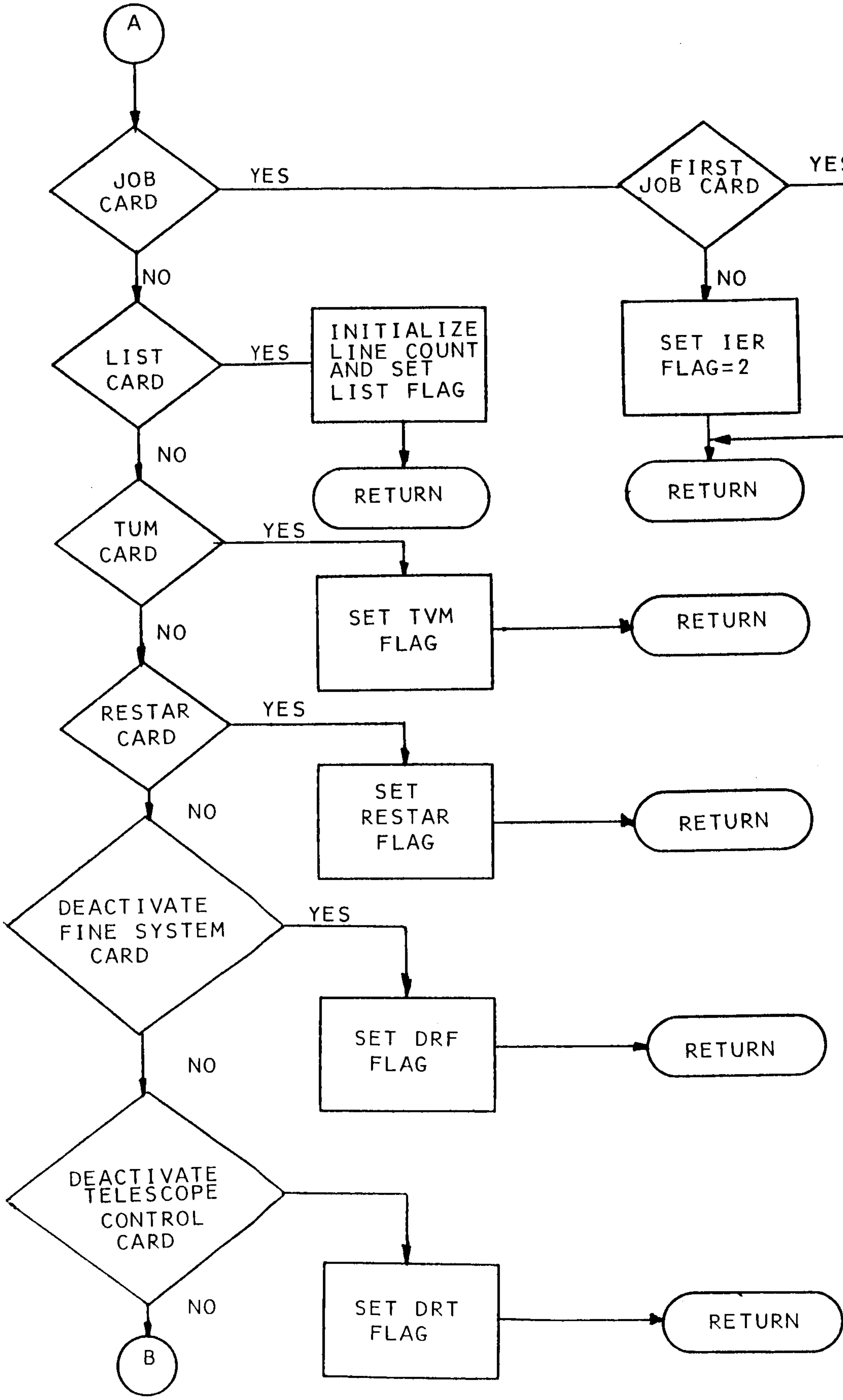
2.3.4 PROCON Subroutine

Subroutine PROCON processes all control cards. Transfer to the appropriate section of instructions is under control of the processing pointer IPOINT which is set in subroutine CHKCRD. For those control cards which require no operand check the appropriate flag is set and return is made to the EXEC subroutine. For other control cards the operand is checked and if valid, the corresponding counter or flag is set. If the operand is found to be invalid, the appropriate error message pointer is set and where possible, corrective action is taken by the program to allow continuation of the run. Otherwise, processing will be terminated after all input has been checked. The function of each control card is described in Table 4-2. The initial processing of print and plot request cards occurs in this subroutine. For both print and plot cards, the requested variables are checked to insure that they are in the allowable list. If so, for plot cards, the data pointer, PLOINT, is set and the X and Y-axis plot labels columns 20-39 and 41-80 respectively are stored on Work Tape 4. For print cards, the print counter, JPRCNT, is increased and the corresponding data pointer PROINT is set.

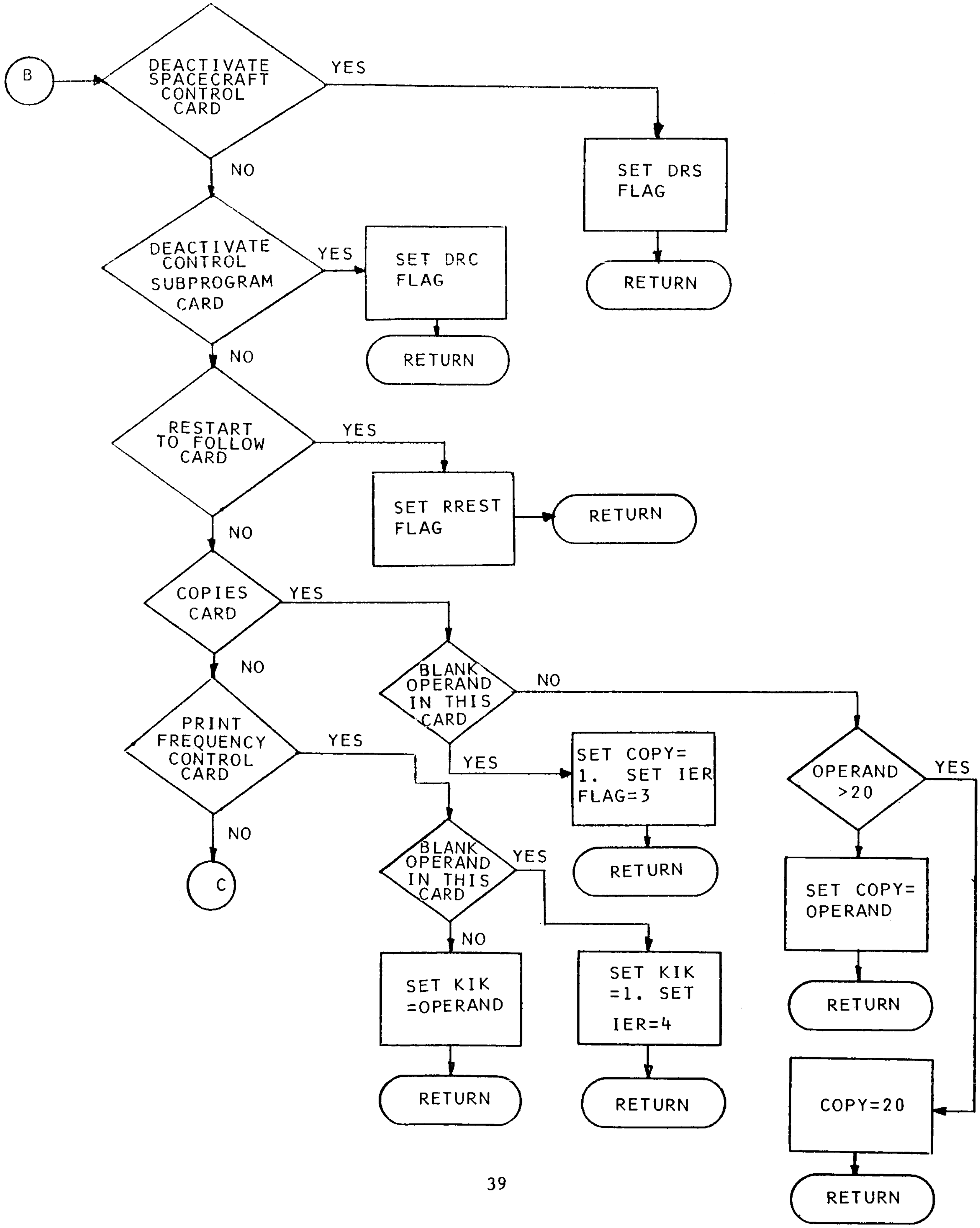
PROCON FLOWCHART



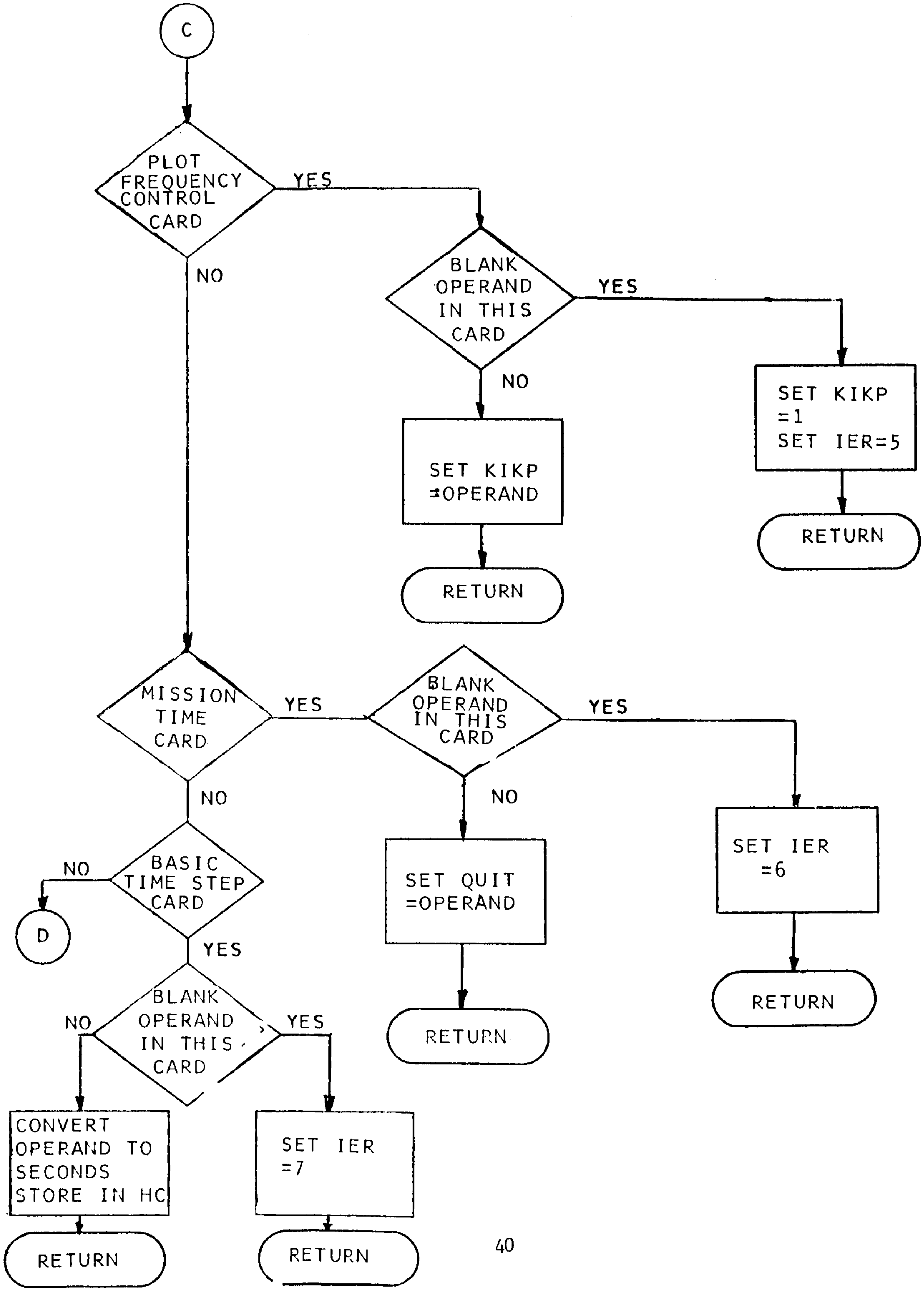
PROCON FLOWCHART (CONTINUED)

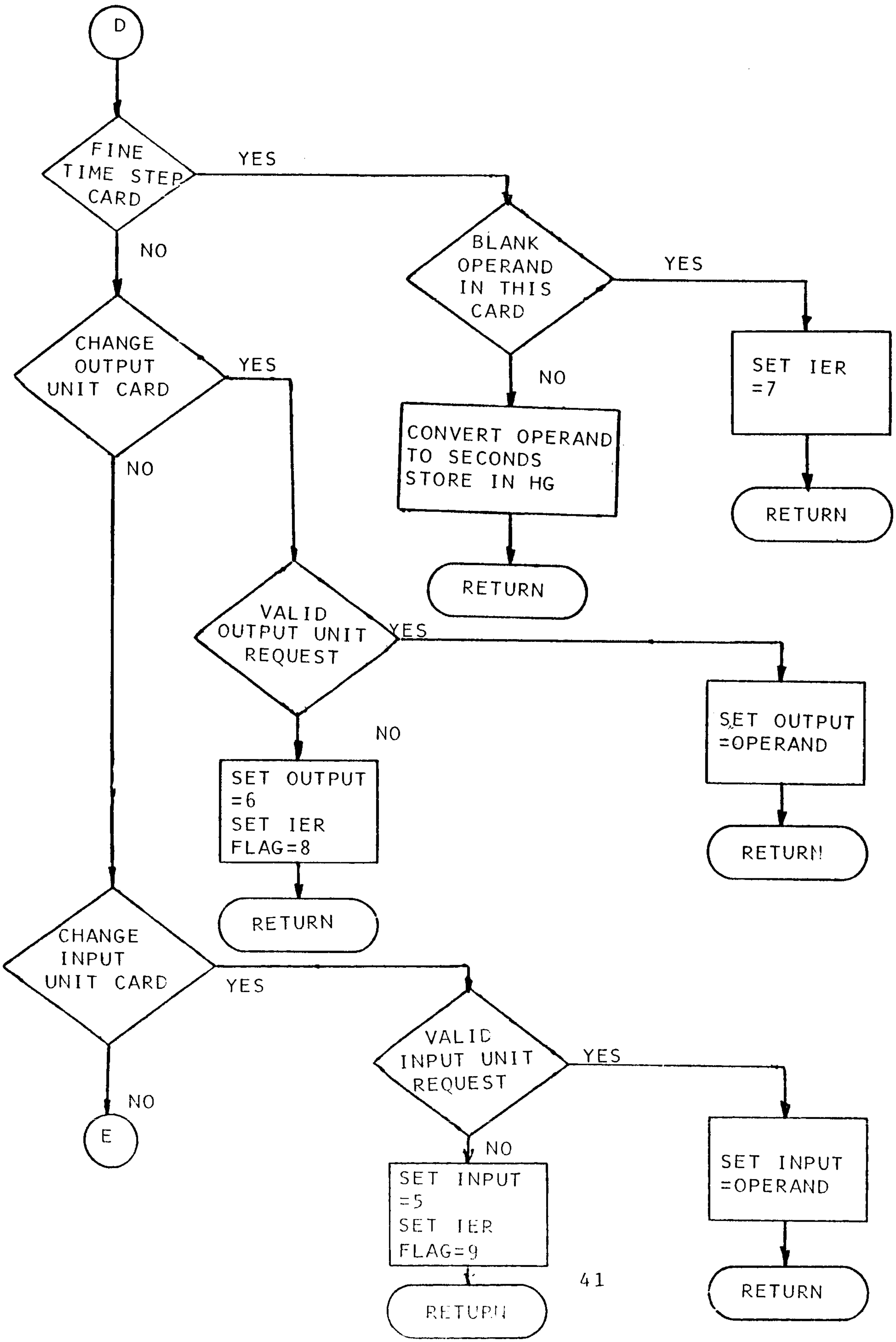


PROCON FLOWCHART (CONTINUED)

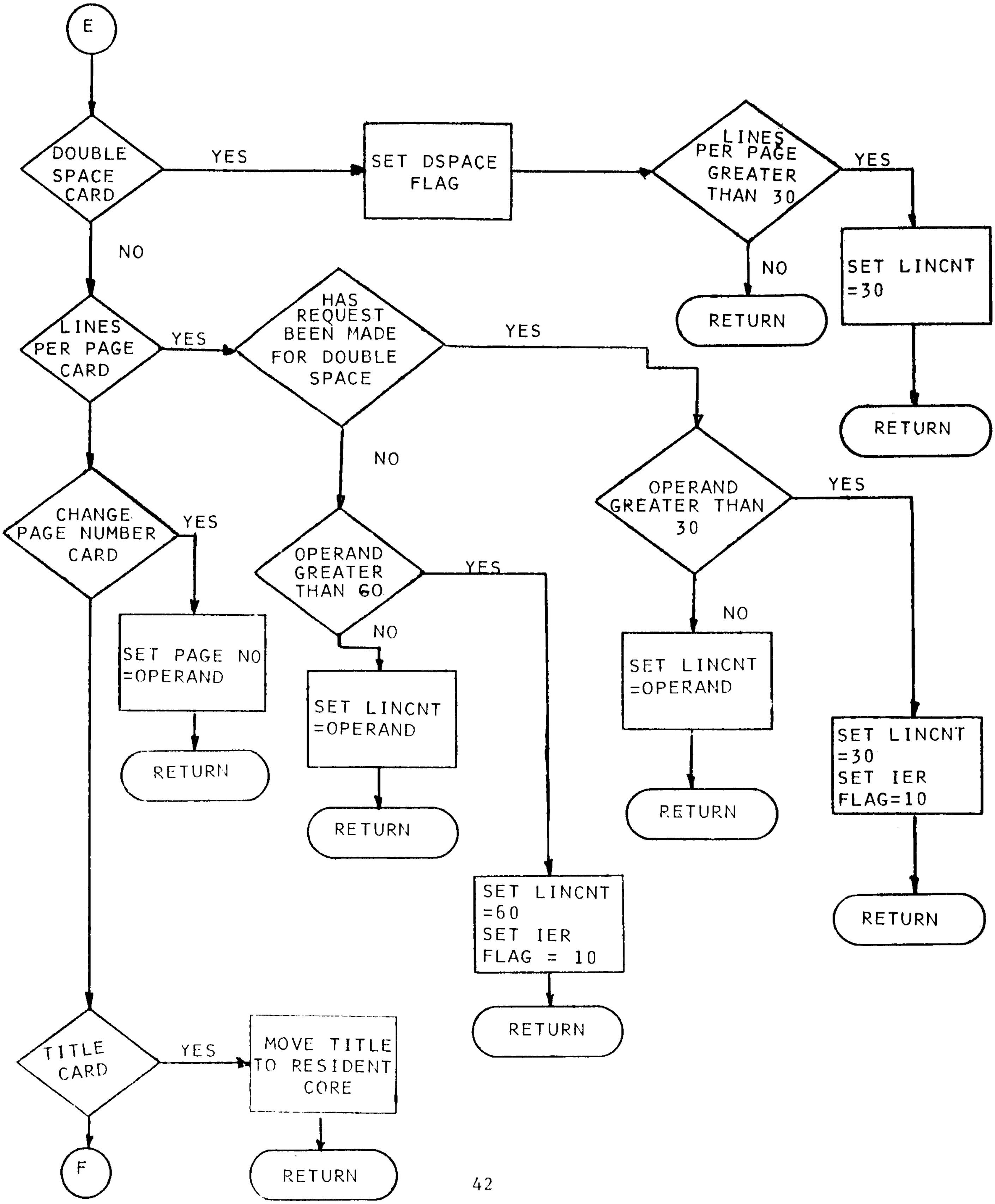


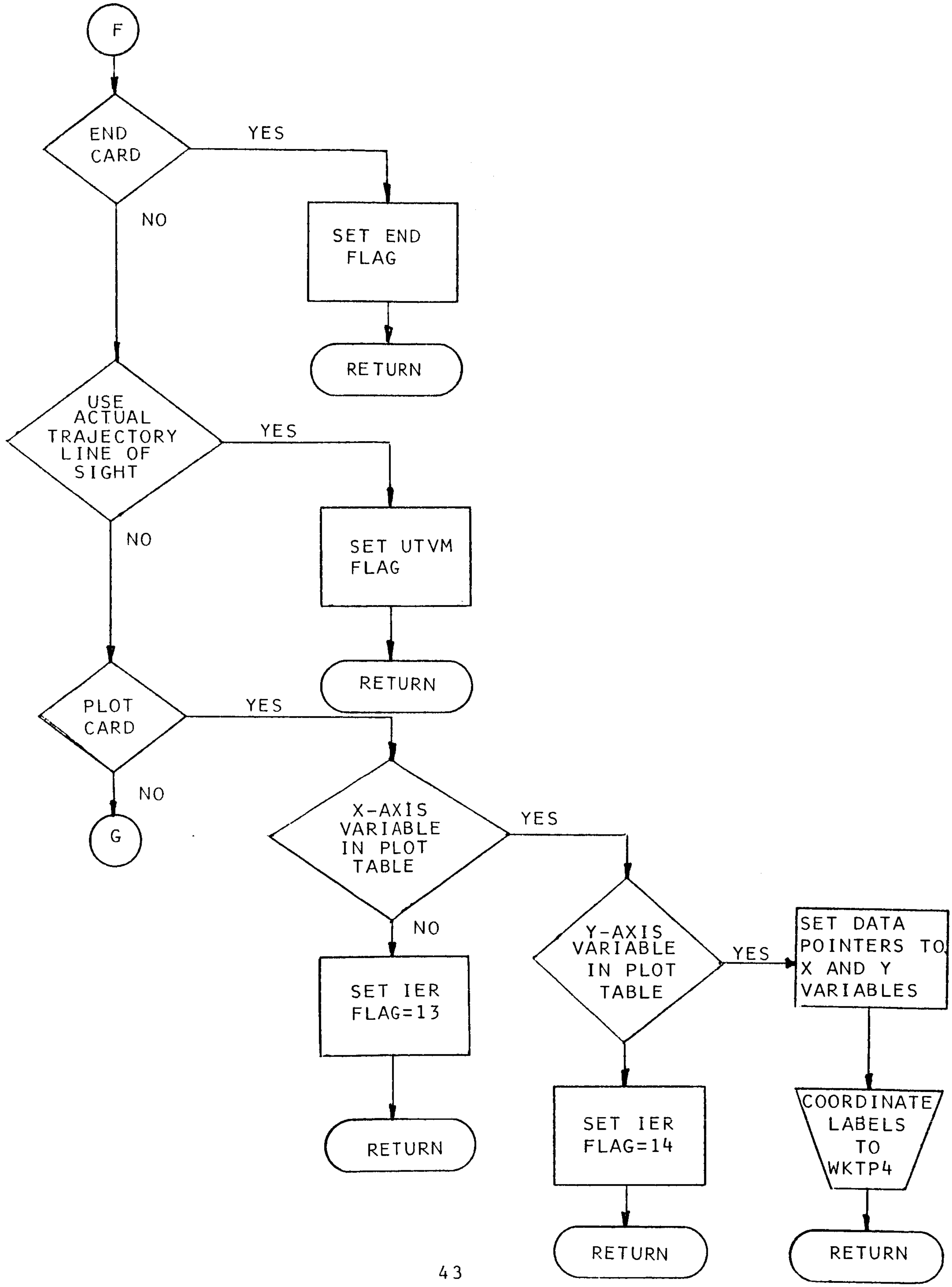
PROCON FLOWCHART (CONTINUED)

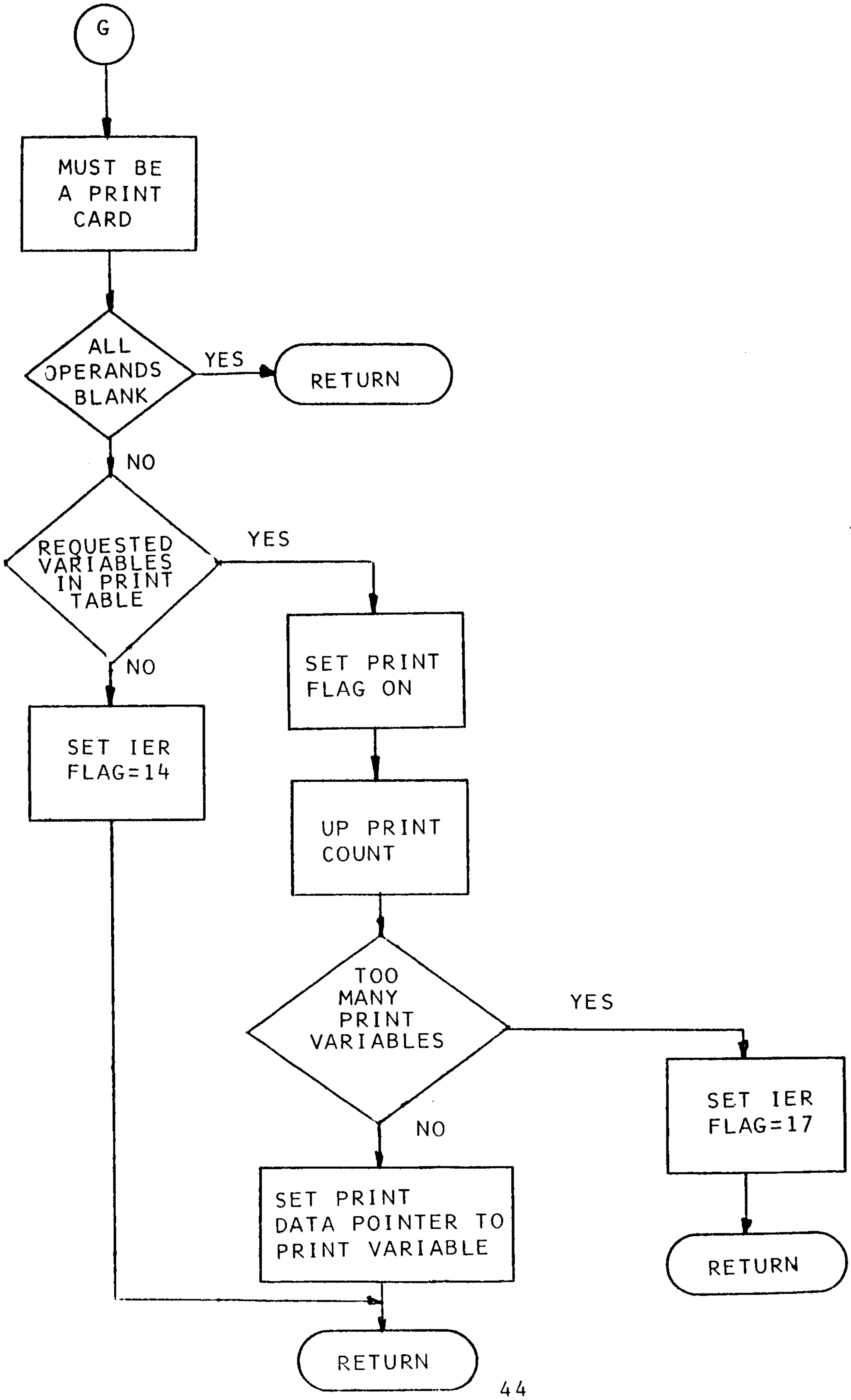



D
FINE TIME STEP CARD
YES
NO
BLANK OPERAND IN THIS CARD
YES
NO
SET IER =7
RETURN
CONVERT OPERAND TO SECONDS STORE IN HG
RETURN
CHANGE OUTPUT UNIT CARD
YES
NO
VALID OUTPUT UNIT REQUEST
YES
NO
SET OUTPUT =OPERAND
RETURN
SET OUTPUT =6 SET IER FLAG=8
RETURN
CHANGE INPUT UNIT CARD
YES
NO
E
VALID INPUT UNIT REQUEST
YES
NO
SET INPUT =OPERAND
RETURN
SET INPUT =5 SET IER FLAG=9
RETURN

PROCON FLOWCHART (CONTINUED)



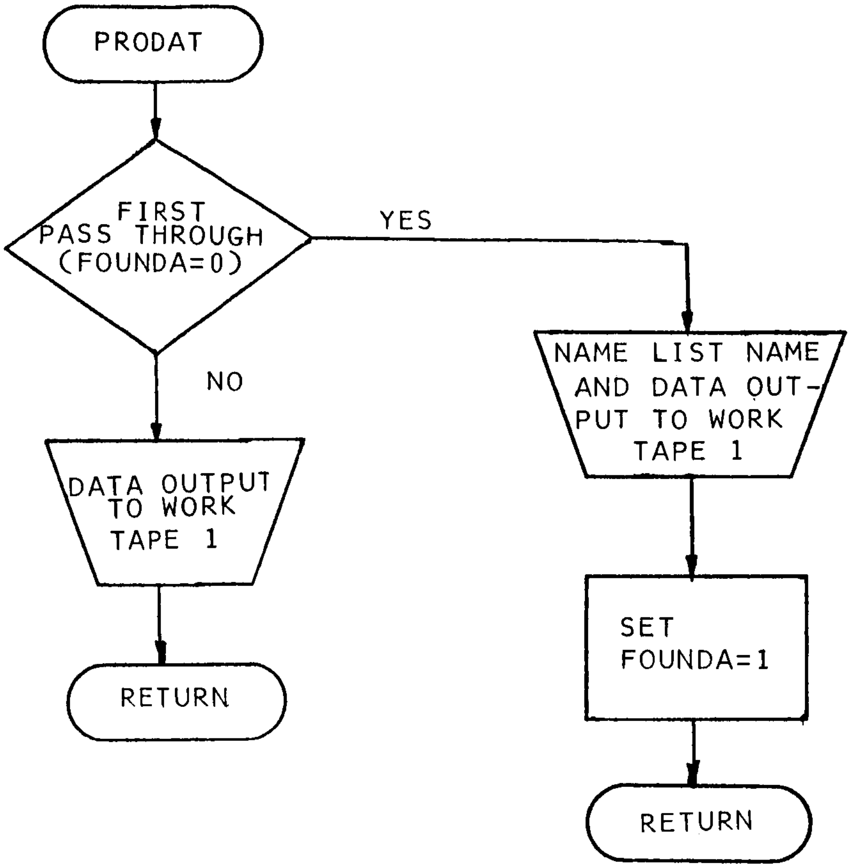

F
END CARD
YES
NO
SET END FLAG
RETURN
USE ACTUAL TRAJECTORY LINE OF SIGHT
YES
NO
SET UTVM FLAG
RETURN
PLOT CARD
YES
NO
G
X-AXIS VARIABLE IN PLOT TABLE
YES
NO
SET IER FLAG=13
RETURN
Y-AXIS VARIABLE IN PLOT TABLE
YES
SET IER FLAG=14
RETURN
SET DATA POINTERS TO X AND Y VARIABLES
COORDINATE LABELS TO WKTP4
RETURN


G
MUST BE A PRINT CARD
ALL OPERANDS BLANK
YES
RETURN
NO
REQUESTED VARIABLES IN PRINT TABLE
YES
NO
SET PRINT FLAG ON
SET IER FLAG=14
UP PRINT COUNT
TOO MANY PRINT VARIABLES
YES
NO
SET IER FLAG=17
SET PRINT DATA POINTER TO PRINT VARIABLE
RETURN
RETURN

2.3.5 PRODAT Subroutine

Subroutine PRODAT creates the data tape (WKTP1) containing user supplied data which will be read in the initialization subroutine (INITR) using Fortran's Namelist feature. On the first pass through this subroutine, the namelist header supplied internal to the subroutine and the user's first data card are written on WKTP1. On succeeding passes only data cards are written. The required namelist end flag ($) is written in subroutine EXEC.

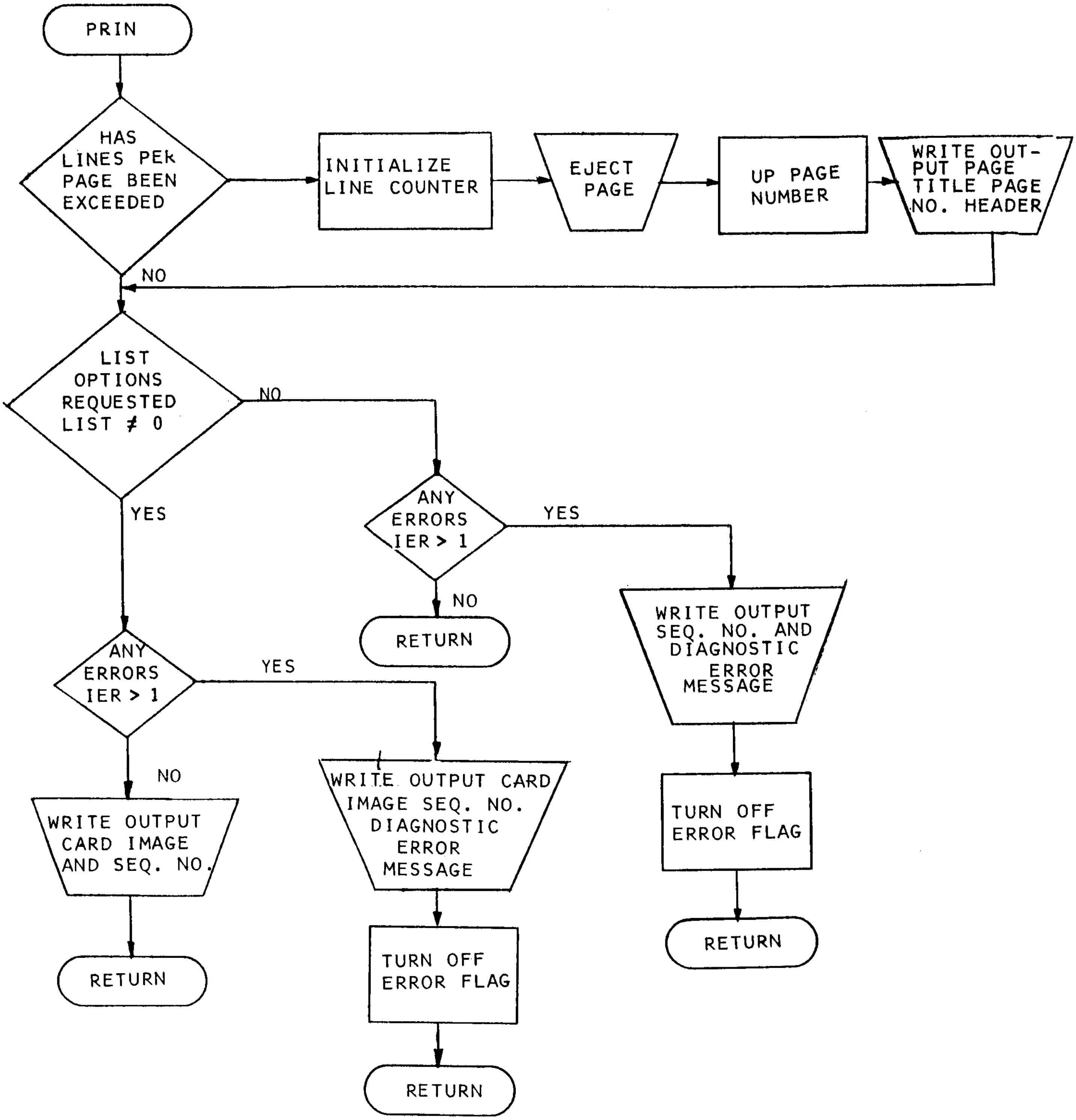
PRODAT FLOWCHART



2.3.6 PRIN Subroutine

Subroutine PRIN writes on the program's output print device the sequence number and diagnostic error message for any invalid user supplied data or control card. These error messages are listed in Table 4-5. If the user has requested a prelist of his input, this subroutine writes the card image of each card supplied by the user following and including the request (the LIST control card). During the processing of an input card, if an error is detected, the error flag IER is set. This flag is used by Subroutine PRIN as a pointer to the appropriate error message. Upon entry to the subroutine, a check is made to see if the current page is full. If so, the page is ejected and the heading information is written at the top of the new page. If the list option has been requested the card image and its sequence number will be written. If an error has been detected in processing the card, the corresponding error message will be written. On completion of the output for a given card, return is made to the calling EXEC subroutine.

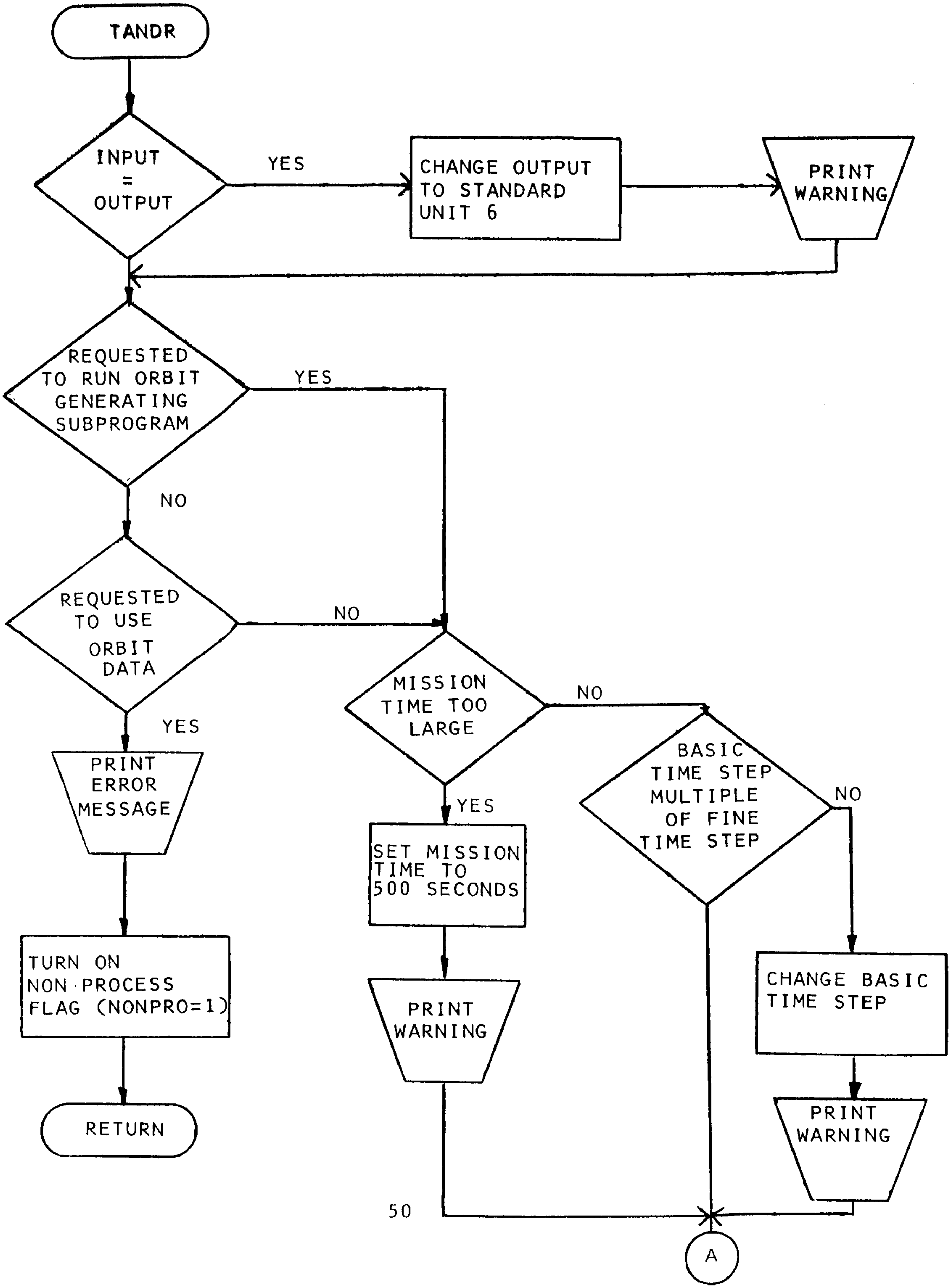
PRIN FLOWCHART

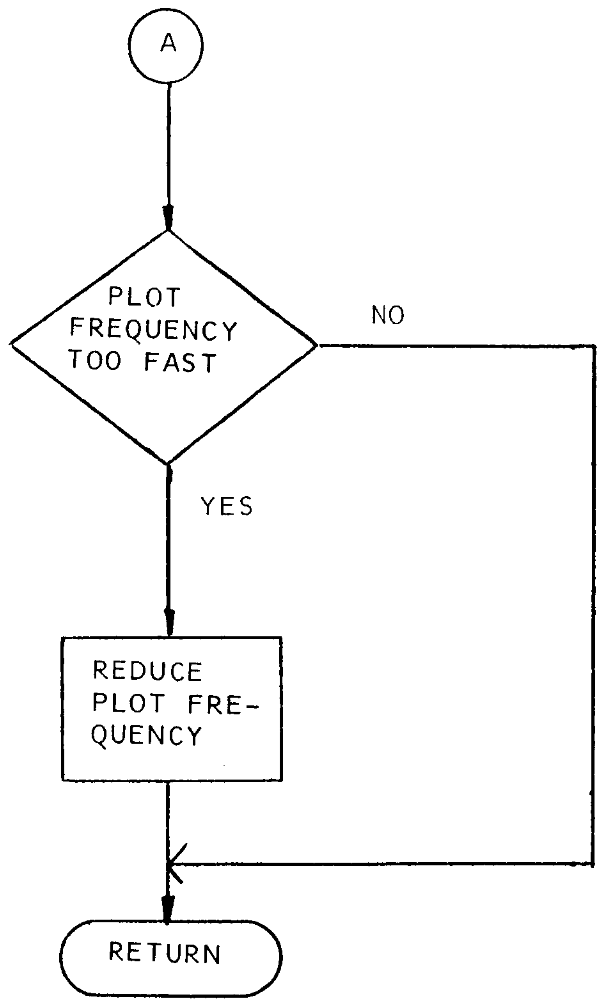


2.3.7 TANDR Subroutine

On entry to subroutine TANDR, a check is made to determine if input and output device requests are the same. If so, the output unit is changed to the standard unit 6. Following this, a check is made to make sure that a request to use orbit generated data is accompanied by a request to generate the data. If not, an error message is printed and the non-process flag is set to cause termination of the job. Return is then made to subroutine EXEC. Otherwise, the program continues with a test to see if mission time is greater than five hundred seconds. If so, it is reduced to five hundred and a warning is printed. A check is then made to insure that the "basic time step" is a multiple of the "fine time step" and if it is not, a warning is printed. Finally, plot frequency is checked and if too fast, it is reduced and the user is warned. Return is then made to subroutine EXEC.

TANDR FLOWCHART

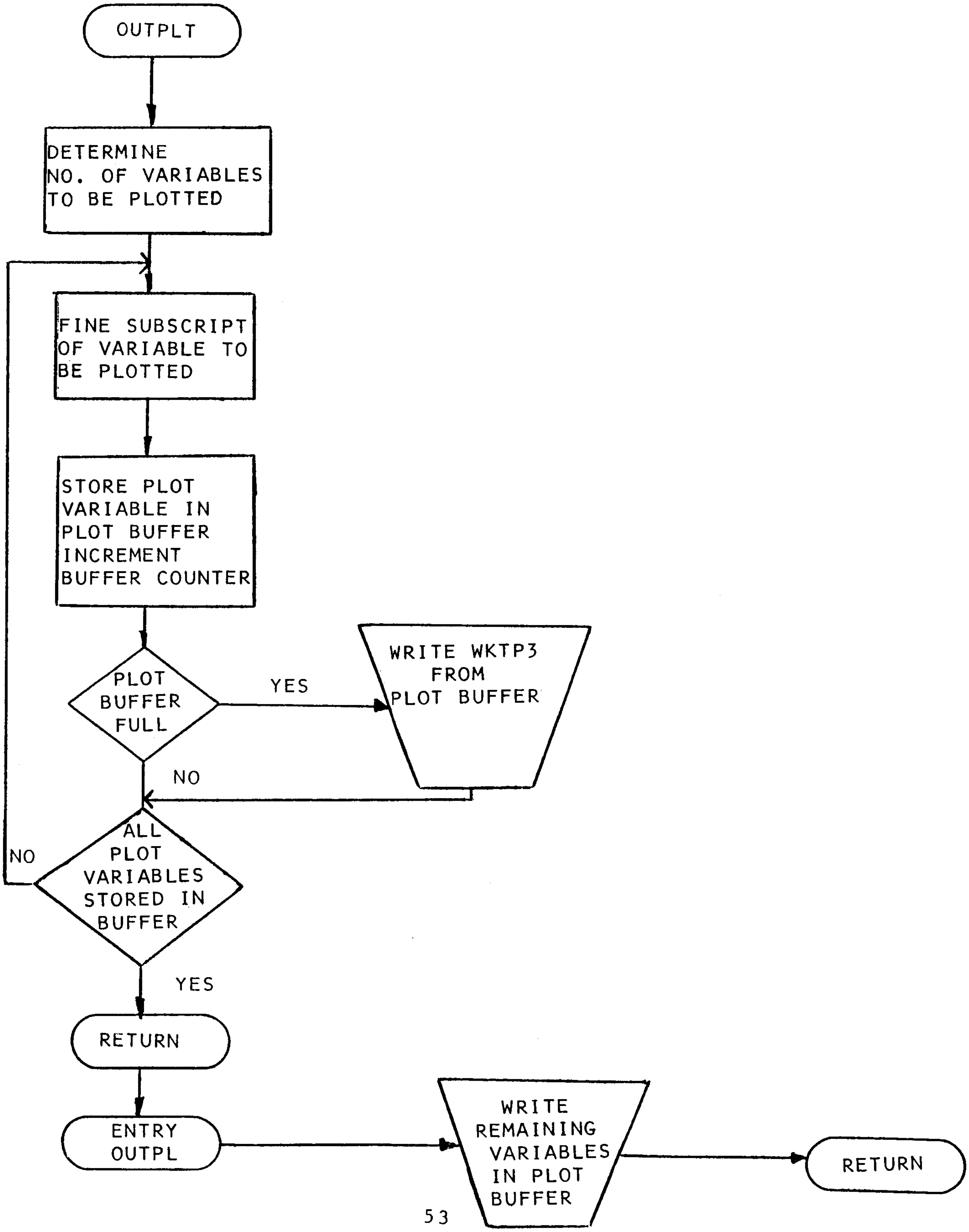



A
PLOT FREQUENCY TOO FAST
NO
YES
REDUCE PLOT FRE- QUENCY
RETURN

2.3.8 OUTPLT Subroutine

Subroutine OUTPLT is invoked throughout the simulation at a frequency which may be specified by the user through control word PLCNTL. Each time the subroutine is called, the variables which the user has requested to be plotted are placed in the plot buffer PLBUF. A check is made after each variable is stored in the buffer to determine if the buffer has reached capacity. When the buffer is filled, its contents are transferred to a work tape, WKTP3. After the final loop through the simulation, the routine will be entered once more to transfer any remaining variables in the buffer to WKTP3. Following execution of this subroutine, return is made to EXEC.

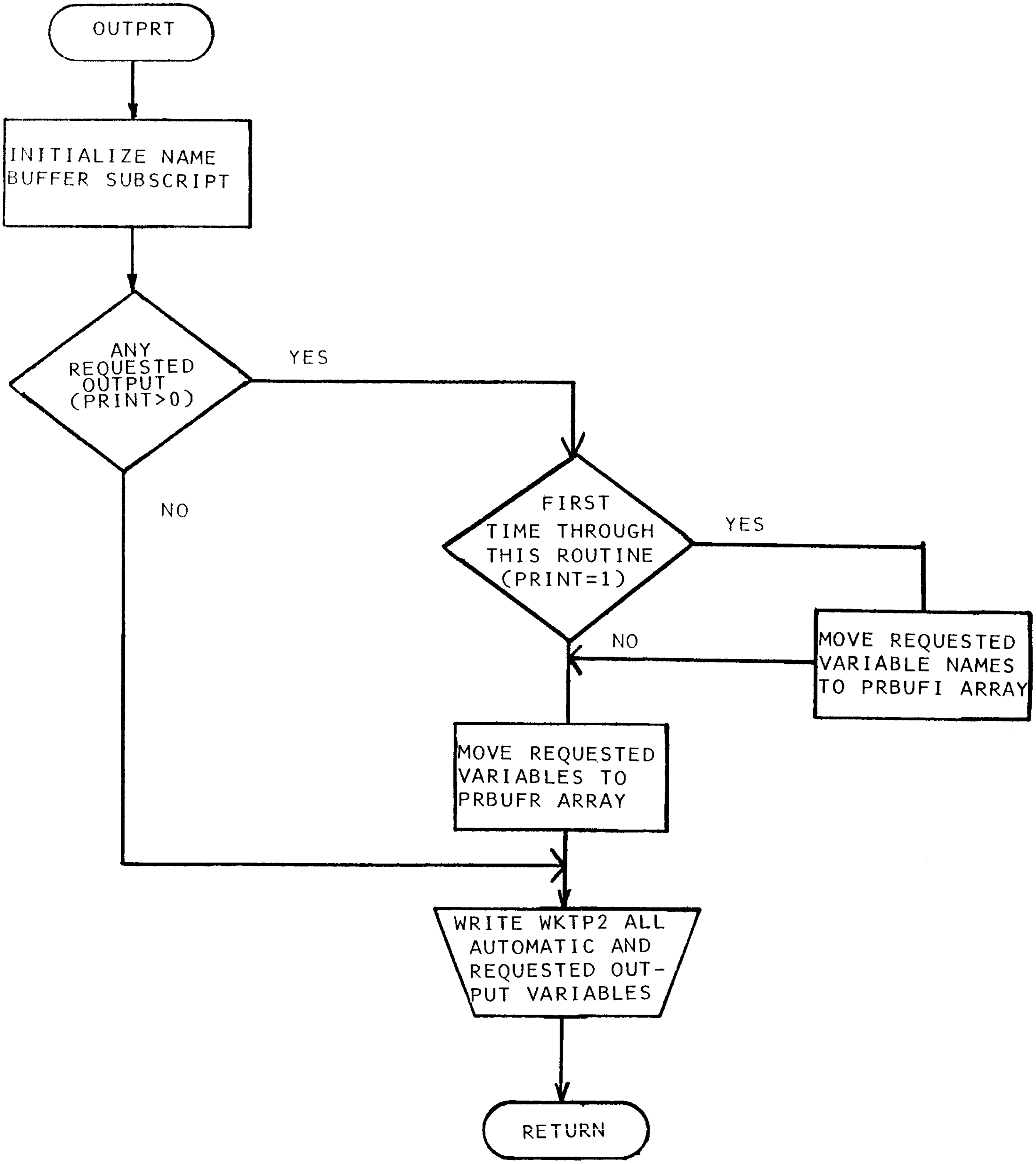
OUTPLT FLOWCHART



2.3.9 OUTPRT Subroutine

Subroutine OUTPRT is used to transfer variables which the user requests to be printed, and those which are automatically printed without request, to work tape WKTP2. OUTPRT is invoked at a frequency which may be supplied by the user through control word PRCNTL. If no request has been received from the user for additional print, PRINT=0, and only those variables which are automatically printed are transferred to WKTP2. If a request has been received to print additional variables, on the first pass through this subroutine, the names of the variables to be printed are stored, including any subscript information, and their values are written on WKTP2. On succeeding passes through this routine PRINT=2 and only the values of the variables to be printed are transferred to WKTP2. When the user specifies the name of a vector or matrix to be printed, he receives output values for each element of the vector or matrix. A two digit subscript not separated by commas and not enclosed in parentheses is generated in this routine for requested matrix output. The first digit represents the matrix row, the second the matrix column. A one digit subscript, not enclosed in parentheses, is generated for requested vector output for each coordinate.

OUTPRT FLOWCHART



2.3.10 OUTPLF Subroutine

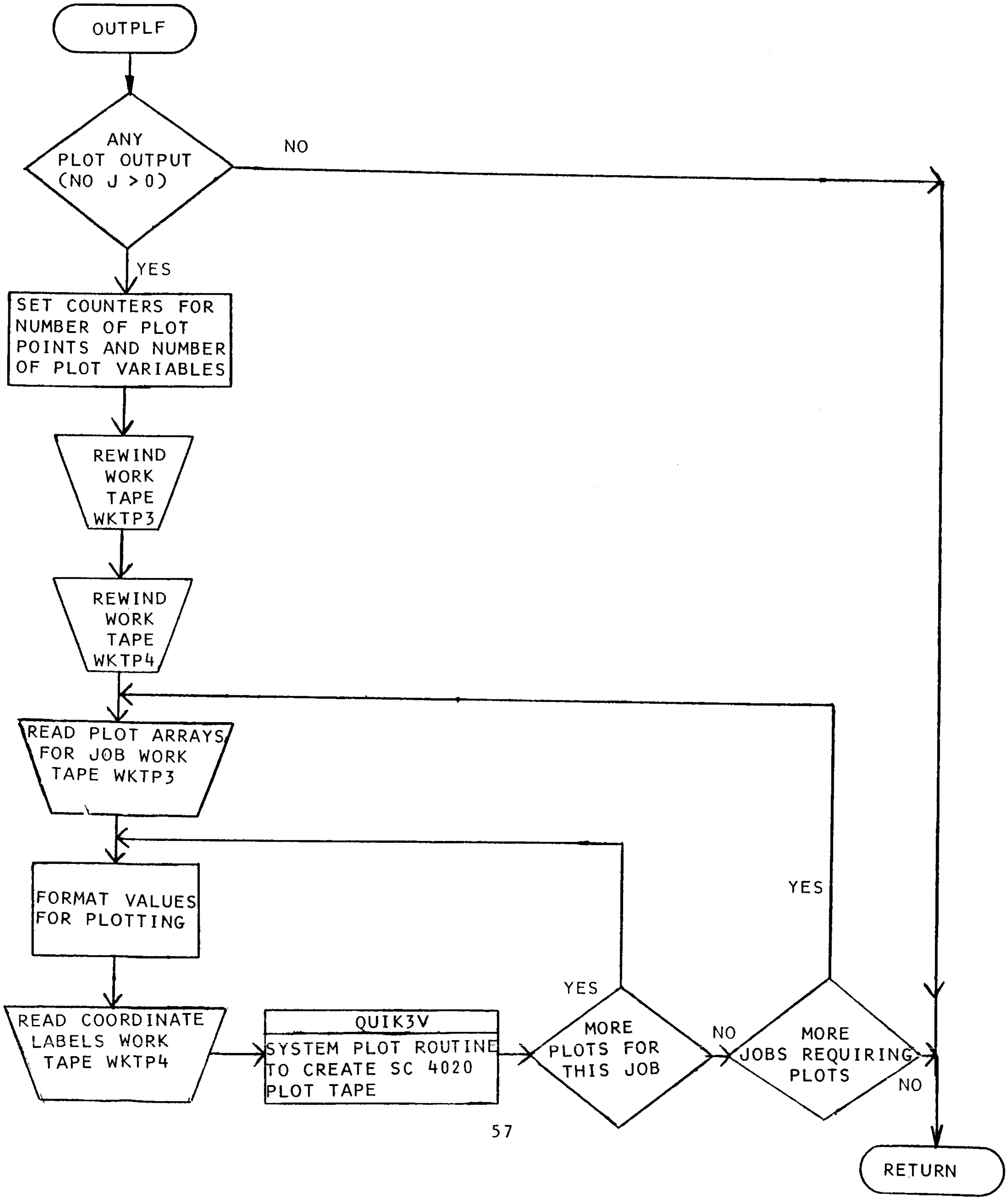
Subroutine OUTPLF serves to read the plot arrays in from tape WKTP3 for all requested plots for all simulation jobs. This routine is executed only after all simulations have been run. Tape WKTP3, which contains the coordinate labels is rewound and read into core. The system plot routine QUIK3V is called which creates the plot tape for the SC 4020. After all plot arrays have been read in and processed by QUIK3V, the program is terminated.

In addition, subroutine OUTPFL performs the processing necessary when logarithmic scaled plots are requested. In OUTPLF, the absolute values of the y coordinates for the log plot are taken and limited to a minimum value of 6×10^{-8} if any fall between zero and this value. This step is required to maintain correct scaling on the log plots for certain variables. The log plots are limited to a range of 10 orders of magnitude for the y coordinate. Thus, if any variables are limited at 6×10^{-8}, the maximum value which will be plotted is 6×10^{2}. The user is advised to consider carefully the range of the variables before requesting log plots. It is to be noted that the absolute value is plotted for any variable for which log scaling is requested.

System subroutine SMXYV is called by OUTPLF to perform the log scaling when required.

The OUTPLF flow chart does not indicate the log scale processing described here. The program listing indicates clearly the processing discussed.

OUTPLF FLOWCHART



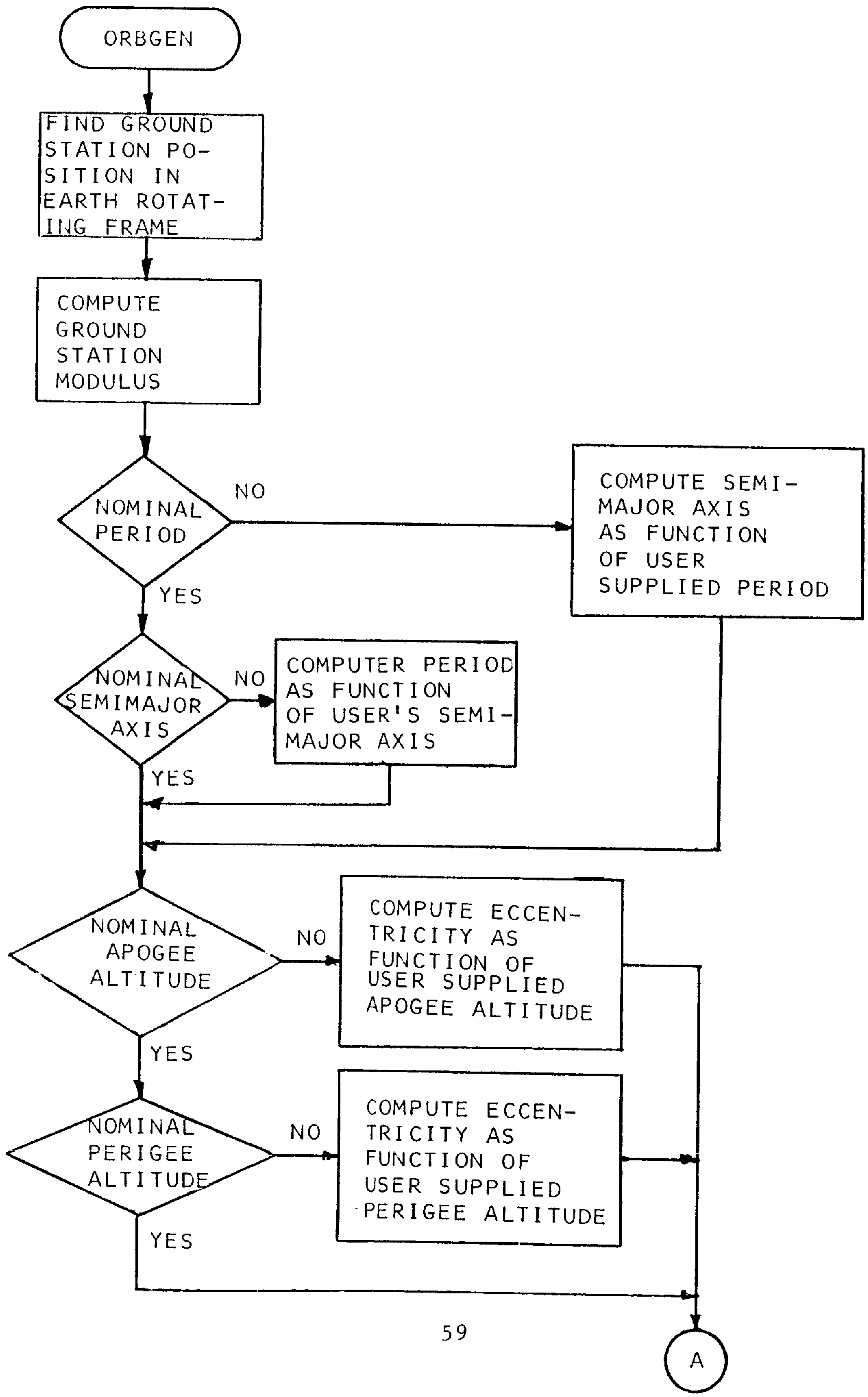
2.3.11 ORBGEN Subroutine

Subroutine ORBGEN is the main subroutine of the Orbit Generating subprogram. It is called only at the request of the user and, if called, it causes an orbit to be generated from nominal parameters or from those supplied by the user. It causes to be stored on magnetic tape and in core, the line-of-sight and line-of-sight rate vectors, computed at a time step equal to one-fiftieth of the mission time. The values computed will be used by the system curve fit subroutine to generate the third order polynomial coefficients, with which the line-of-sight and line-of-sight rate are computed at the high frequency required by the simulation. The user may specify the orbit through the input of orbital direction through insertion, insertion latitude, argument of perigee, inclination of orbit, semi-major axis or period, eccentricity or apogee altitude or perigee altitude, and insertion longitude. If the semi-major axis is input, the program computes the orbit period. If apogee altitude or perigee altitude is input, the program computes the eccentricity. The user will also input the location of the ground station. If no input is received for this program but use of it is requested, a nominal orbit will be generated and the line-of-sight will be provided from points in this orbit to the fixed ground station.

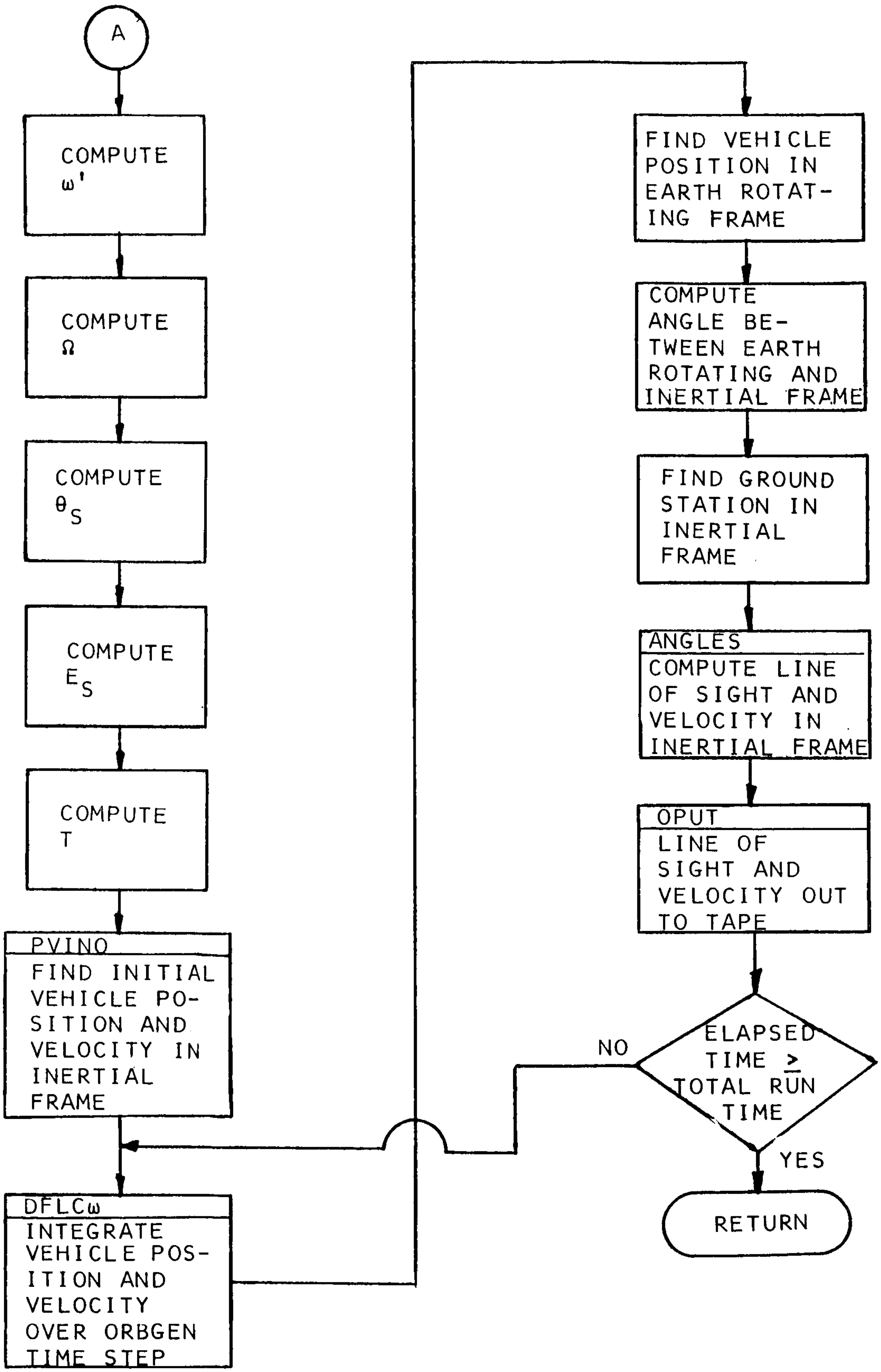
If ORBGEN is invoked, the location of the ground station is found in the earth's rotating reference frame. If the user has input the period, the orbit's semi-major axis is computed and if the user has input the semi-major axis, the period is computed. The program next determines if the user has supplied either apogee altitude or perigee altitude. If so, the eccentricity is computed as a function of the altitude. The program then computes the argument of perifocus, longitude of ascending node, true anomaly, eccentric anomaly, and time of perifocal passage. Following these computations, the initial position and velocity of the vehicle is determined in the inertial frame by subroutine PVINO. Entry is then made to subroutine DLFCW to set up parameters required for continued integration. Succeeding calls to DFLCW will result in the integration of the vehicle's position and velocity over the ORBGEN time step. This integration is accomplished using Cowell's technique and employs subroutine DERIV to supply the solutions to the orbital equations of motion. Next, the ground station position in the inertial frame is calculated so that the line-of-sight vector from ground station to vehicle and its velocity may be computed. A call to subroutine ANGLES follows where the line-of-sight vector and its velocity are computed in the inertial frame.

Subroutine OPUT is then called, where the line-of-sight vector and its velocity are then output to tape. If, at this point in the logic, the orbit generation elapsed time equals or exceeds the specified time, a return is made to EXEC. If not, execution will resume at the call to subroutine DFLCW to integrate over another time step.

ORBGEN FLOW


ORBGEN
FIND GROUND STATION POSITION IN EARTH ROTATING FRAME
COMPUTE GROUND STATION MODULUS
NOMINAL PERIOD
NO
YES
COMPUTE SEMI-MAJOR AXIS AS FUNCTION OF USER SUPPLIED PERIOD
NOMINAL SEMIMAJOR AXIS
NO
YES
COMPUTER PERIOD AS FUNCTION OF USER'S SEMI-MAJOR AXIS
NOMINAL APOGEE ALTITUDE
NO
YES
COMPUTE ECCENTRICITY AS FUNCTION OF USER SUPPLIED APOGEE ALTITUDE
NOMINAL PERIGEE ALTITUDE
NO
YES
COMPUTE ECCENTRICITY AS FUNCTION OF USER SUPPLIED PERIGEE ALTITUDE
A

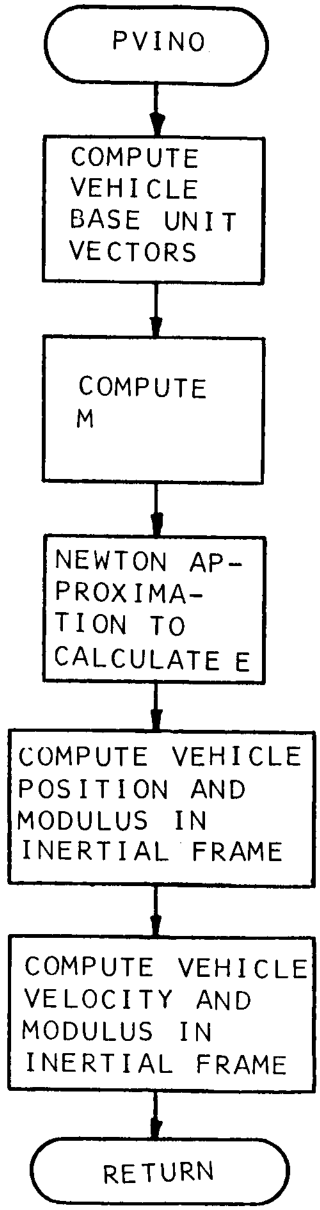
ORBGEN FLOW
(CON'T)



2.3.12 PVINO Subroutine

Subroutine PVINO is called to determine the initial position and velocity of the spacecraft. On entry to the routine the vehicle base unit vectors are computed. Then, the mean anomaly, M, is calculated. If $M > 0$, an initial estimate of the eccentric anomaly, E, is taken as an odd multiple of π radians nearest M. Succeeding estimates of E are computed via Newton's approximation method. When the approximations converge or after the twenty-fifth iteration, or in the event $M = 0$, the x and y coordinates of the spacecraft relative to the vehicle base unit vector triad are computed. The position and velocity of the vehicle relative to inertial space is then computed and return is made to subroutine ORBGEN.

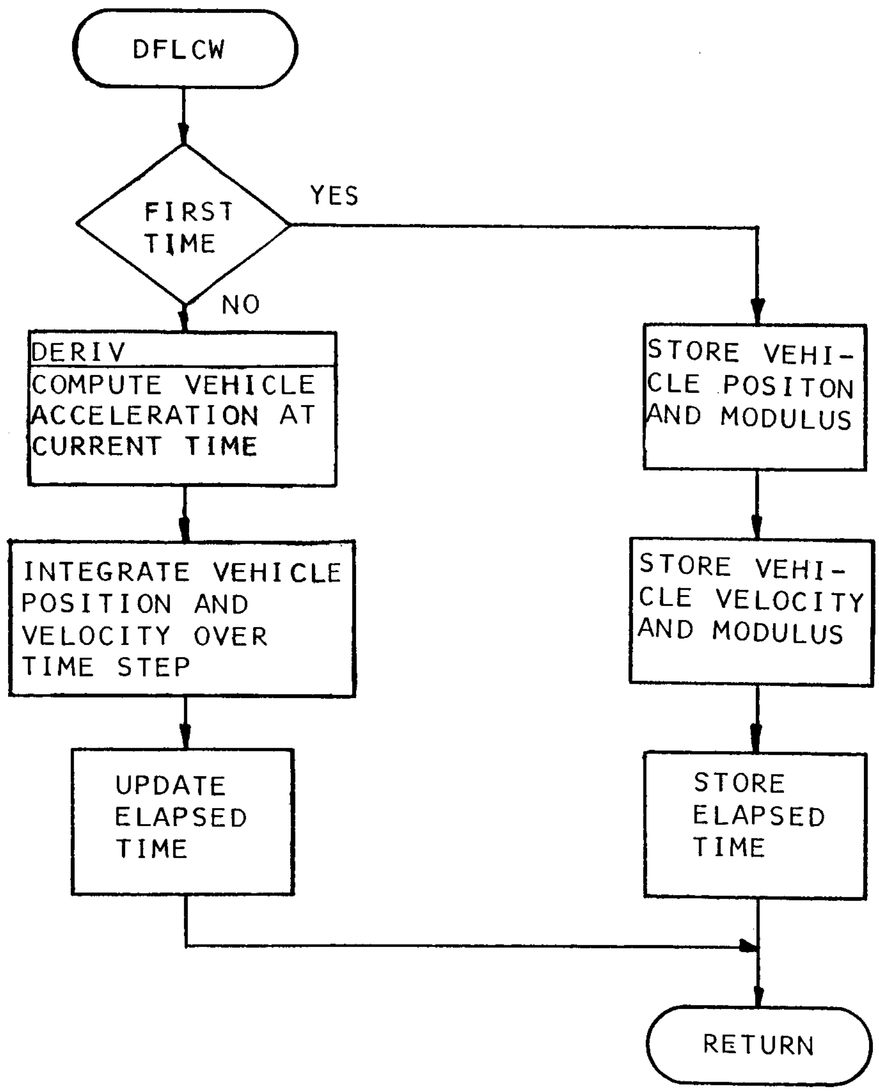
PVINO FLOWCHART



2.3.13 DFLCW Subroutine

On entry to subroutine DFLCW, the flag INIT is checked. If it is zero, it indicates that the subroutine is being called for the first time and the vehicle position and velocity and elapsed time are stored. The routine then returns control to subroutine ORBGEN. If INIT is non-zero, subroutine DERIV is called for the computation of spacecraft acceleration. Updated values for vehicle position and velocity are obtained through Cowell Integration. Elapsed time is updated and control is returned to subroutine ORBGEN.

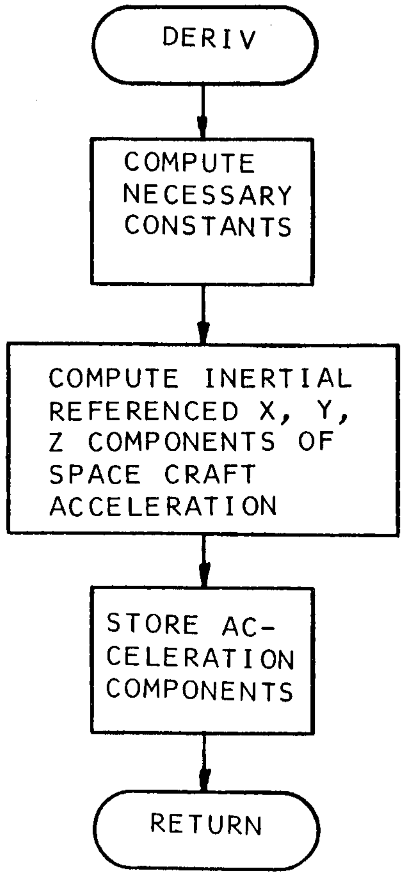
DFLCW FLOWCHART



2.3.14 DERIV Subroutine

Subroutine DERIV computes components of spacecraft acceleration due to the earth's gravitational potential.

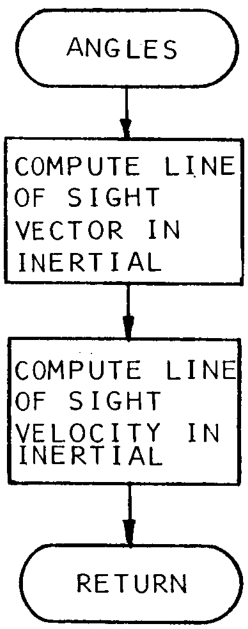
DERIV FLOWCHART



2.3.15 ANGLES Subroutine

Subroutine ANGLES is called to obtain the line-of-sight and line-of-sight rate vectors for each orbit generating time step. Taking the difference of the x, y, z components of the spacecraft position and ground station in the inertial frame, the subroutine obtains the line-of-sight vector. Similarly, the difference of the components of spacecraft velocity and ground station velocity yield the line-of-sight rate vector.

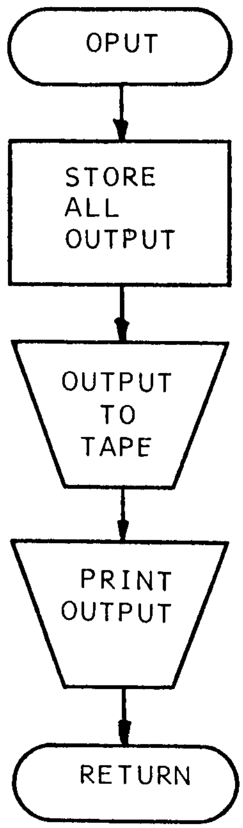
ANGLES FLOWCHART



2.3.16 OPUT Subroutine

For each time step in the Orbit Generating subprogram, subroutine OPUT is called to record on tape and to print the output created during the previous pass. After this routine functions, return is made to subroutine ORBGEN.

OPUT FLOWCHART



2.3.17 FINE Subroutine

Subroutine FINE is the main subroutine of the Fine Tracking subprogram. In addition to determining the logic flow through the subprogram, it is used to compute the transfer lens velocity. When subroutine FINE is entered, two tests are made to determine if the ground beacon is or has been in the fine field-of-view. First, the Z component of the normalized line-of-sight vector in the telescope frame, SD3, is tested to see if it is greater than the cosine of 1 arc minute. If the answer is yes, the beacon is in the fine field-of-view and program word FLAG is set equal to one. FLAG is initialized zero and will remain so until this test is passed. If SD3 is less than the cosine of 1 arc minute, the second test is made to see if FLAG is zero. If not, this indicates the beacon is no longer in the fine field-of-view but was previously and FLAG is set to two.

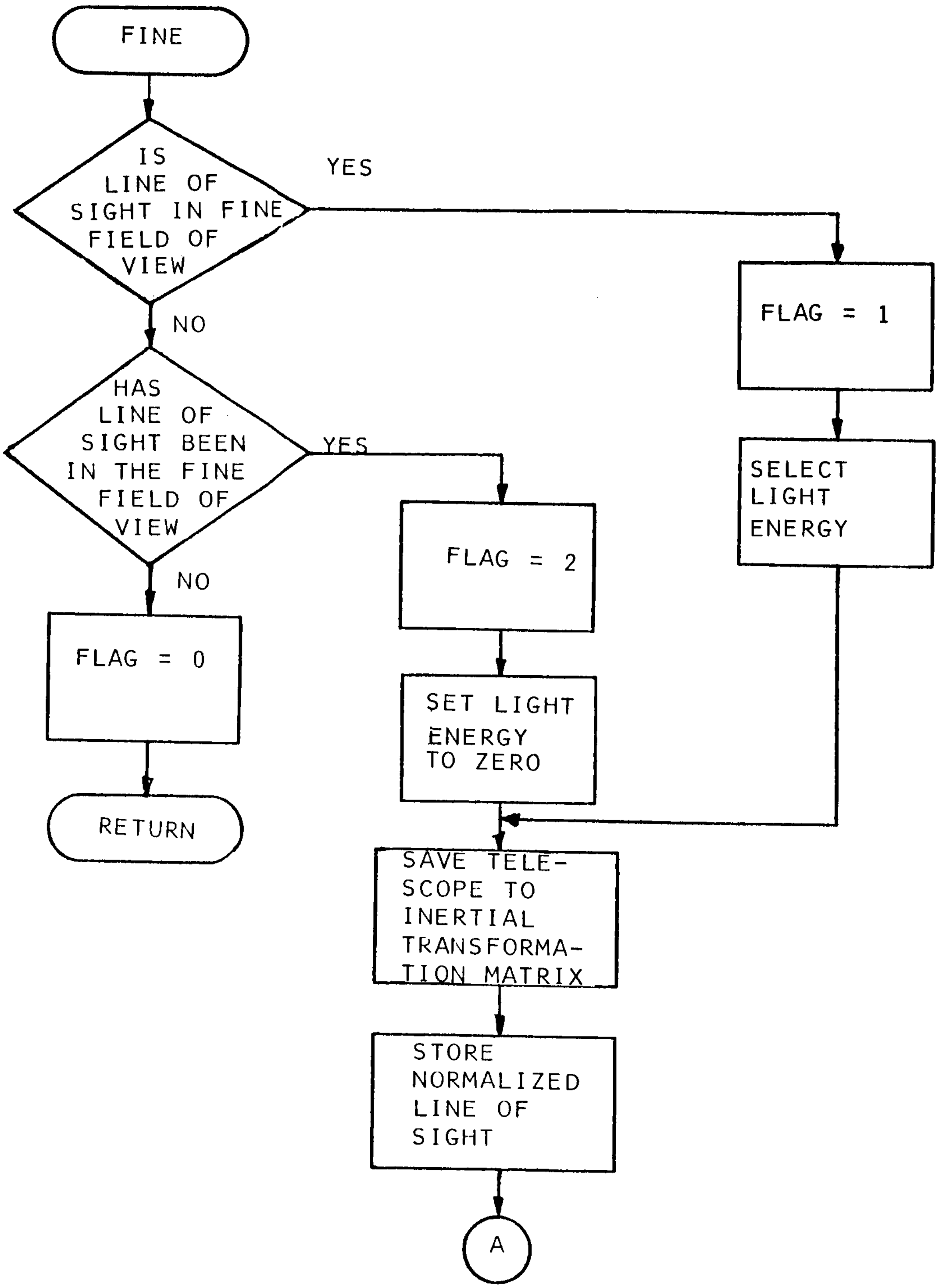
If the ground beacon is not nor has not been in the fine field-of-view, FLAG remains zero and control returns to EXEC. If the beacon is in the fine field-of-view, the total light energy is set to the appropriate value and the fine tracking loop entered. If the beacon is not now in the fine field but was previously, the total light energy is set equal to zero and the fine tracking loop entered.

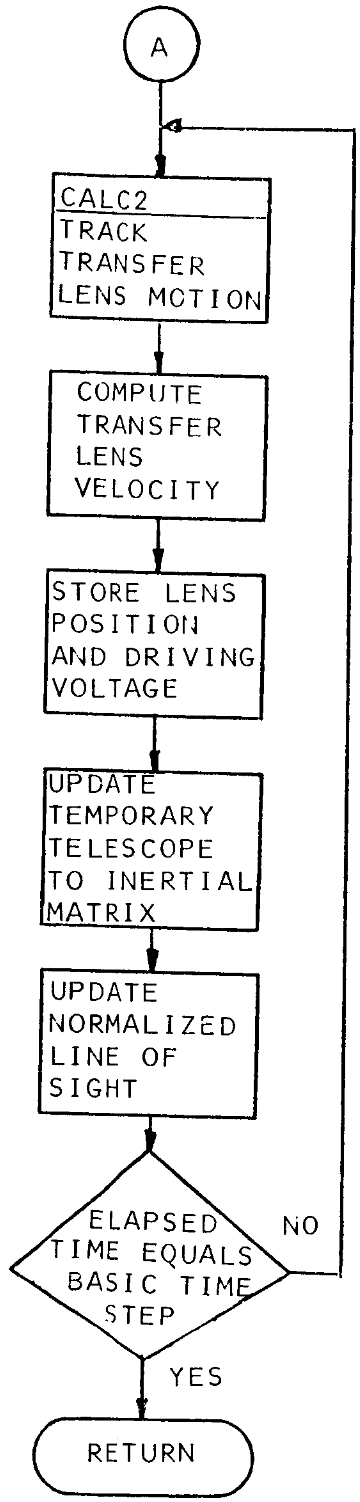
With a decision to enter the fine tracking loop, the matrix is first set equal to T2I. This initializes the line-of-sight updating process which must be performed after each pass through the fine tracking loop.

In the fine tracking loop, subroutine CALC2 is called to simulate the fine tracking system hardware and compute a new transfer lens position. When this is accomplished control returns to FINE. Though not used as an input to any control system, the transfer lens velocity is then calculated in FINE. Next, the [T2I] matrix element rates are used to extrapolate new values for the [T2I] matrix elements over the fine tracking time step. This updated matrix is then used to update the normalized line-of-sight vector components in the telescope frame. Components of this vector are used as error signals in simulating the fine tracking system for the next pass.

This process is repeated in the fine tracking loop until the elapsed time in the loop is equal to the basic time step, at which time control transfers back to EXEC.

FINE FLOWCHART




A
CALC2
TRACK TRANSFER LENS MOTION
COMPUTE TRANSFER LENS VELOCITY
STORE LENS POSITION AND DRIVING VOLTAGE
UPDATE TEMPORARY TELESCOPE TO INERTIAL MATRIX
UPDATE NORMALIZED LINE OF SIGHT
ELAPSED TIME EQUALS BASIC TIME STEP
NO
YES
RETURN

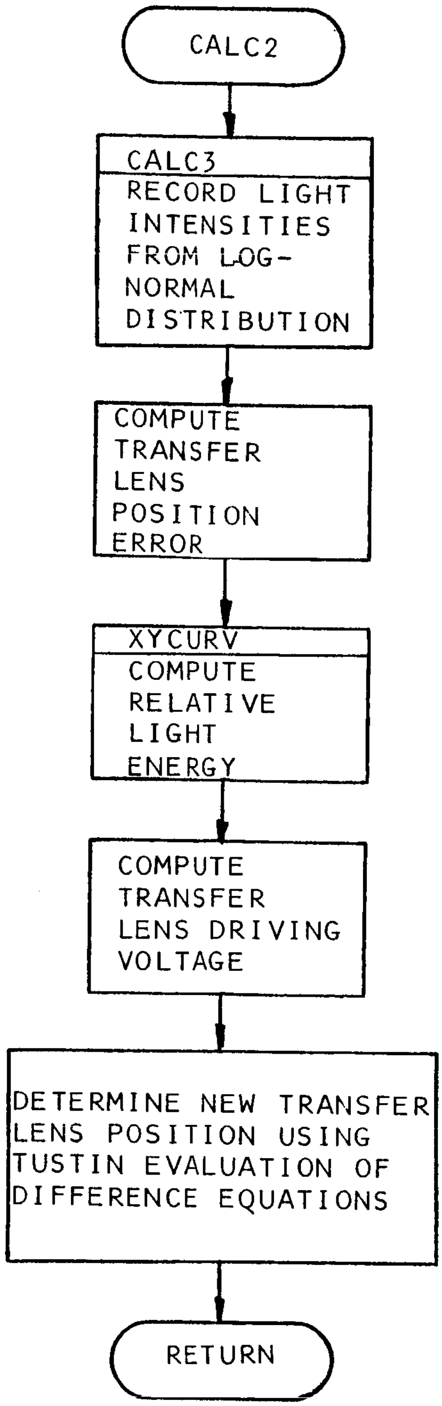
2.3.18 CALC2 Subroutine

Subroutine CALC2 simulates fine tracking system operation. On entry to this subroutine a call is made to subroutine CALC3 where light energies from a log-normal distribution are determined. Following this, the image position in the f/70 focal plane is computed. Then subroutine XYCURV is called where image position is used to compute energy fractions which in turn are used to compute the transfer lens driving voltage. Then, difference equations representing control system hardware are evaluated. From this, the new transfer lens position is obtained and a return is then made to subroutine FINE.

As presently programmed, a constant value for total light energy falling on the f/70 focal plane is used in place of the value computed in CALC3. In CALC3 and subroutines INTENS and RNG, a value is computed for the total light energy which is a random variable. The set of random energies so computed has an amplitude distribution which is log-normal. However, power density spectrum weighting has not been provided for the random variable. Adequate definition of the expected power density spectrum or auto correlation function was not available and development of the random energy generation is incomplete. The mechanism for generating the random energy amplitudes has been left in the program in the form of subroutines CALC3, INTENS, and RNG so that only adding the power density spectrum weighting need be accomplished to generate a randomly varying total light energy with realistic values.

Once definition of the required power density spectrum is obtained, a digital network may be synthesized to modify the amplitudes presently generated to reproduce the desired energy.

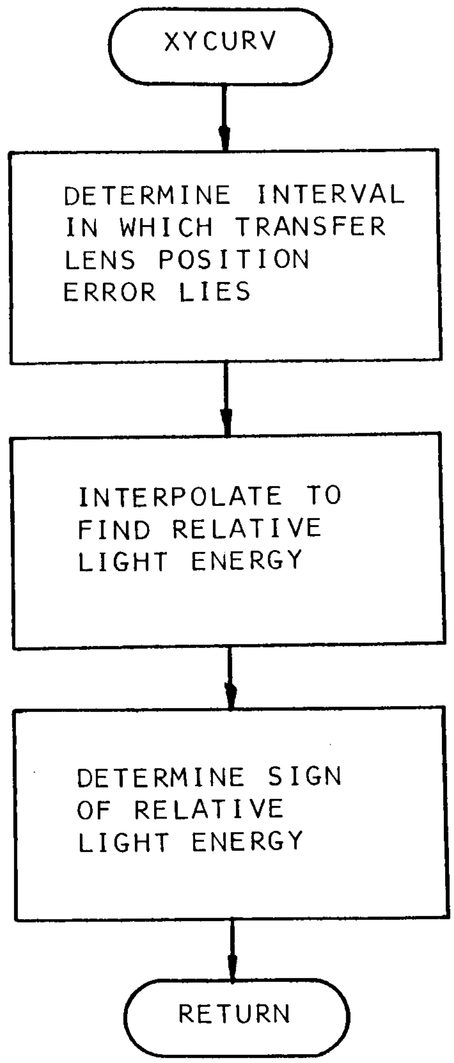
CALC2 FLOWCHART



2.3.19 XYCURV Subroutine

Stored in subroutine XYCURV is a function representing the energy fraction falling on one side of a knife edge boundary as a function of image position coordinates. On entry to subroutine XYCURV, image position coordinate values are scanned to determine between which two lies the current coordinate, computed in subroutine CALC2. An interpolation is then made between the corresponding two energy fraction values to obtain an energy fraction corresponding to the current image coordinate. The sign of the light energy is then determined and return is made to subroutine CALC2.

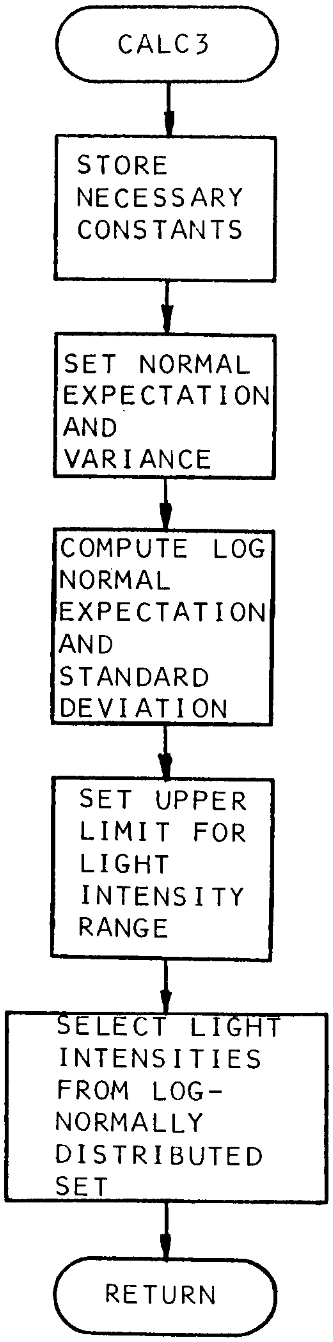
XYCURV FLOWCHART



2.3.20 CALC3 Subroutine

When subroutine CALC3 is entered, all necessary constants for the generation of light energy with a log normal distribution are stored. A log-normal Expectation and Variance are then computed in terms of a normal Expectation and Variance. An upper limit is then set for the range from which light energies with a log-normal distribution will be selected. Using a standard statistical approximation, a set of light energies is obtained as a function of the log-normal Expectation and Variance.

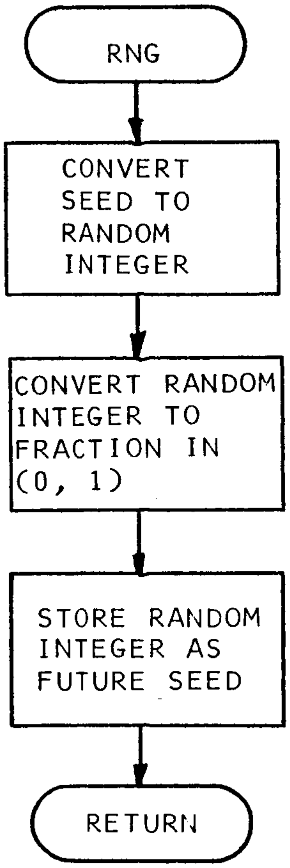
CALC3 FLOWCHART



2.3.21 RNG Subroutine

Subroutine RNG is used to generate a random number each time it is called. Using an integer seed as the generator, the seed is converted to a random integer. The integer is then converted to a fraction in the unit interval [0,1]. Following this, the integer is stored to be used as the seed when the routine is next called. Return is then made to the subroutine. This subroutine is not called as presently programmed.

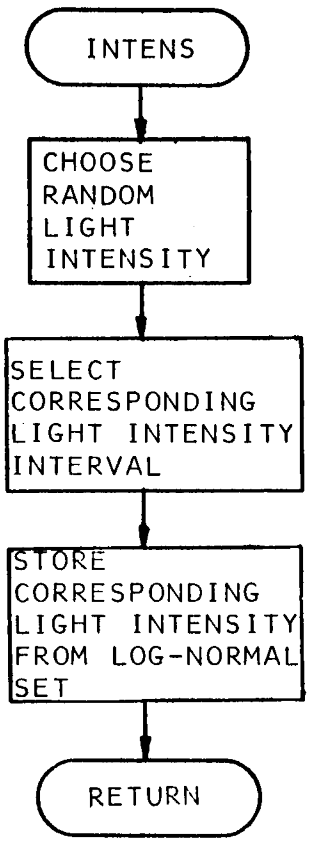
RNG FLOWCHART



2.3.22 INTENS Subroutine

On entry to subroutine INTENS a random number is used to choose a random light energy in the range determined in subroutine CALC3. The corresponding light energy interval in which the random light energy lies is found. Then, the light energy from the log-normally distributed set is chosen which corresponds to the interval. This light energy is stored and return is made To the calling subroutine. This subroutine is not called as presently programmed.

INTENS FLOWCHART



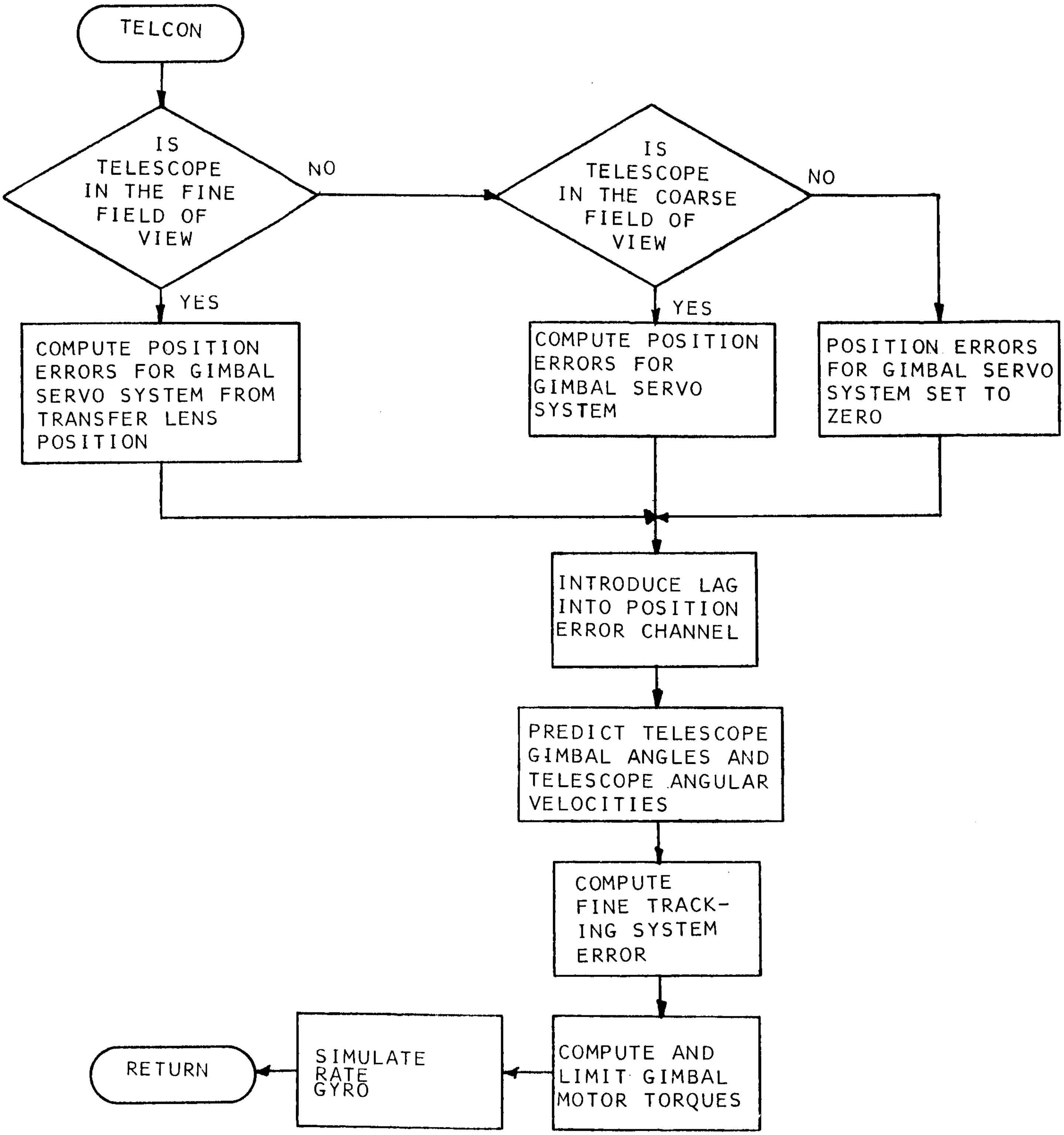
2.3.23 TELCON Subroutine

On entry to subroutine TELCON a check is made to determine if the ground beacon is in the fine field-of-view. If FLAG=1, the beacon is in the fine field and the position errors for the gimbal control system are computed as a function of transfer lens position. If FLAG $\neq$ 1, a check is made to determine if the telescope is in the coarse field of view. If so, the position errors for the gimbal control system are set and the correct sign is attached; otherwise, the position errors are set to zero. Lag is then introduced in the position error channel through evaluation of a first order lag difference equation. Next, telescope gimbal angles and telescope angular velocities are predicted ahead by trapezoidal integration.

The difference equations representing the rate gyros are evaluated using the predicted telescope rates to determine the velocity loop damping signals for control. These signals are combined with the position error signals in the calculation of the motor torques. These torques are limited to the saturation values programmed.

Finally, the quantities EPX and EPY, representing total tracking system error are computed and a return is made to EXEC.

TELCON FLOWCHART

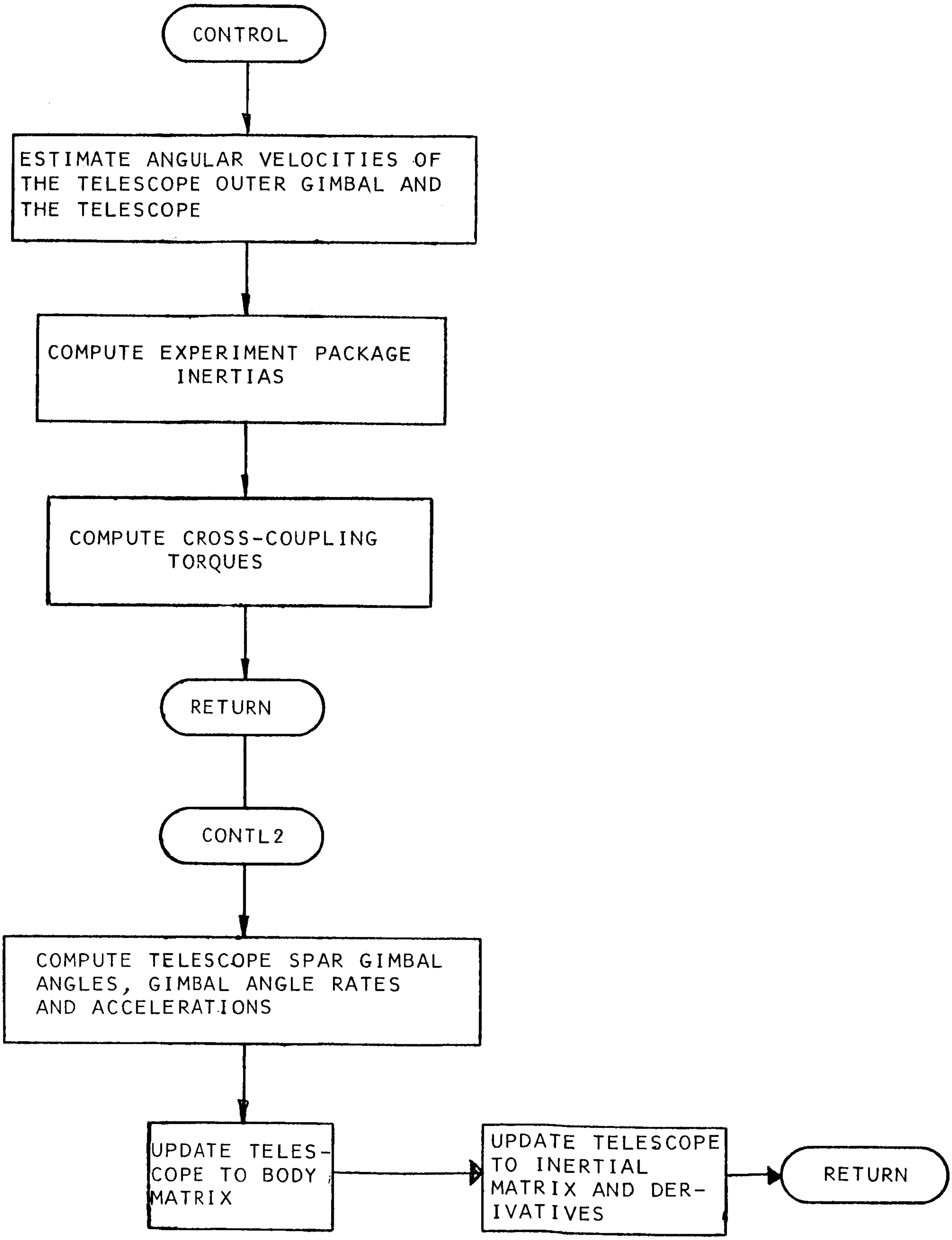


2.3.24 CONTRL Subroutine

Subroutine CONTRL has two entry points; the first subroutine entry is called CONTRL and the second entry point is called CONTL2. On entry to subroutine CONTRL, the angular velocities of the telescope and spacecraft are predicted ahead one basic time step as are the gimbal angles. These predictions are then used in computing experiment package inertias. Next, cross coupling torques which act along the inner and outer gimbal axes are computed and control is returned to subroutine EXEC. These functions are performed before execution of the spacecraft attitude control subprogram. On entry to CONTL2, after execution of the spacecraft attitude control subprogram, telescope angular accelerations are computed and the gimbal angle rates and angles updated. The new values may be thought of as revised values for those predicted in CONTRL.

The telescope-to-body, [T2B], transformation matrix is then updated. Finally, subroutine DIRCOS is called to update the telescope to inertial transformation, [T2I], matrix and the derivatives of the matrix elements. Return is then made from CONTL2 to subroutine EXEC.

CONTRL FLOWCHART



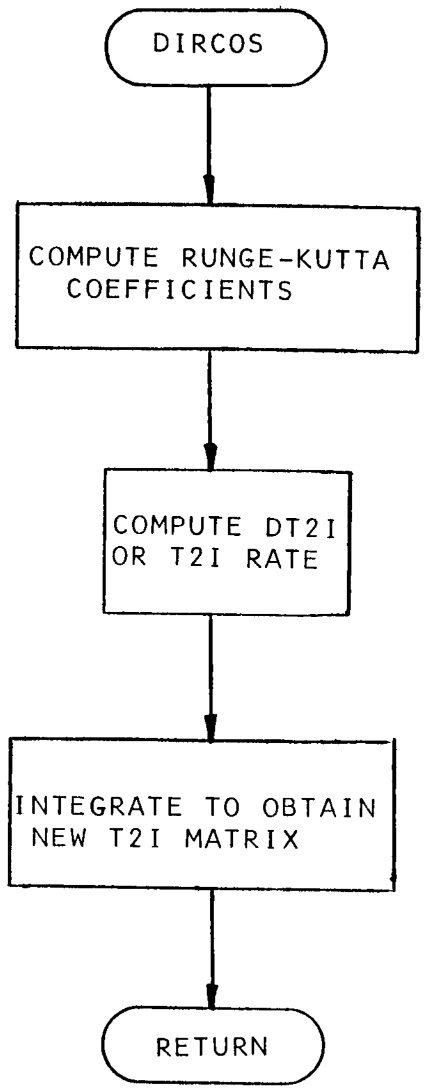
2.3.25 DIRCOS Subroutine

The DIRCOS subroutine is called by the CONTL2 subroutine once per computation cycle. The DIRCOS subroutine updates the telescope to inertial or T2I matrix and also the rates of the elements of this matrix.

The basic inputs to DIRCOS are the angular velocity vector components of the telescope frame relative to inertial space. In the subroutine, equations are solved for the T2I matrix element rates or the DT2I matrix in terms of the angular velocity components of the telescope frame. These equations are then integrated using a fourth order Runge-Kutta method to obtain the new T2I matrix.

The T2I matrix is calculated in DIRCOS every basic time step (.01 sec). DT2I or the matrix of T2I element rates is used in the FINE subprogram to predict a value for the T2I matrix at .002 second or the fine time step intervals between major computation cycles. Upon completion of the subroutine, control is returned to CONTL2.

DIRCOS FLOWCHART

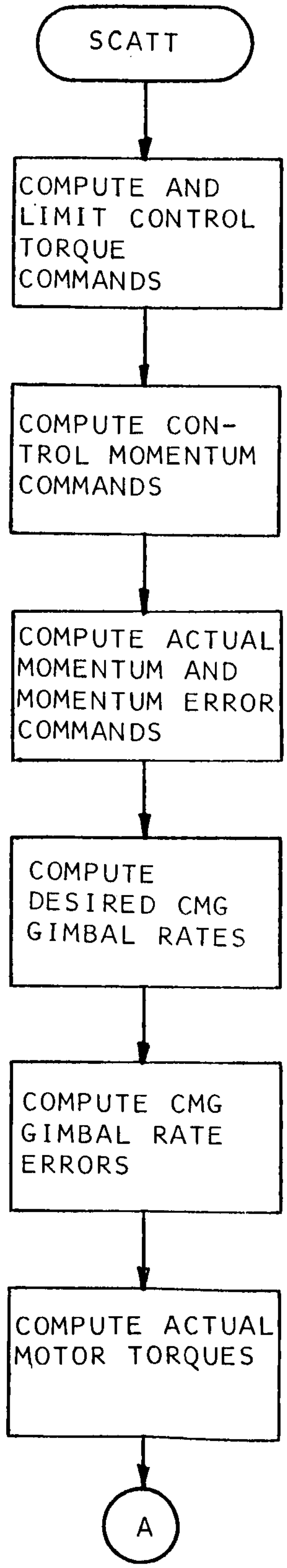


2.3.26 SCATT Subroutine

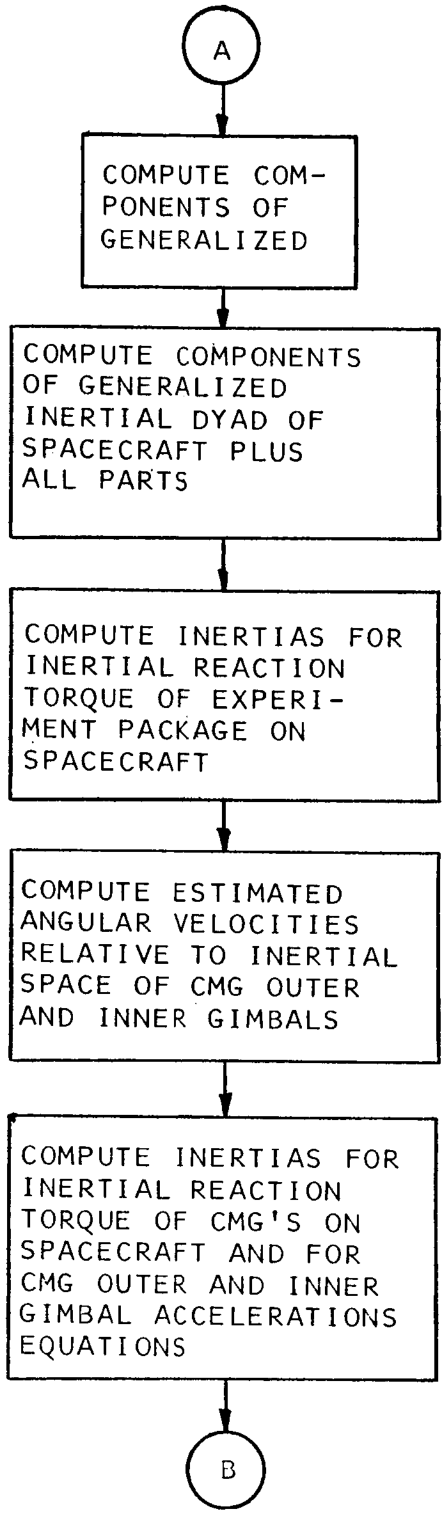
Subroutine SCATT consists of a series of computations which result in the solutions of the CMG control system equations, the spacecraft attitude dynamics equations, and the CMG dynamics equations. On entry to subroutine SCATT, torque commands and control momentum commands are computed. Then the actual momentum and momentum error commands are computed. Next, the CMG gimbal rate commands are computed from the momentum error commands and the H-vector control law matrix. The CMG gimbal servo loops are simulated using the rate commands as input. Difference equations are solved to simulate the control hardware, the end result being gimbal motor torques.

Spacecraft and experiment package inertias are then computed and the angular velocities of the CMG outer and inner gimbals are estimated. The next sequence of computations leads to the determination of net torque on the spacec raft. In these, the inertias for the inertial reaction torque of CMG's on the spacecraft are first computed. Then, cross-coupling torques are computed. A call to subroutine ASTROM is made to provide external torques acting on the spacecraft due to astronaut motion. Then, the net torque on the spacecraft is computed. The angular acceleration and angular velocity of the spacecraft are then computed. Finally, the CMG outer and inner gimbal angle accelerations, velocities and positions are computed. A return to subroutine EXEC is then made.

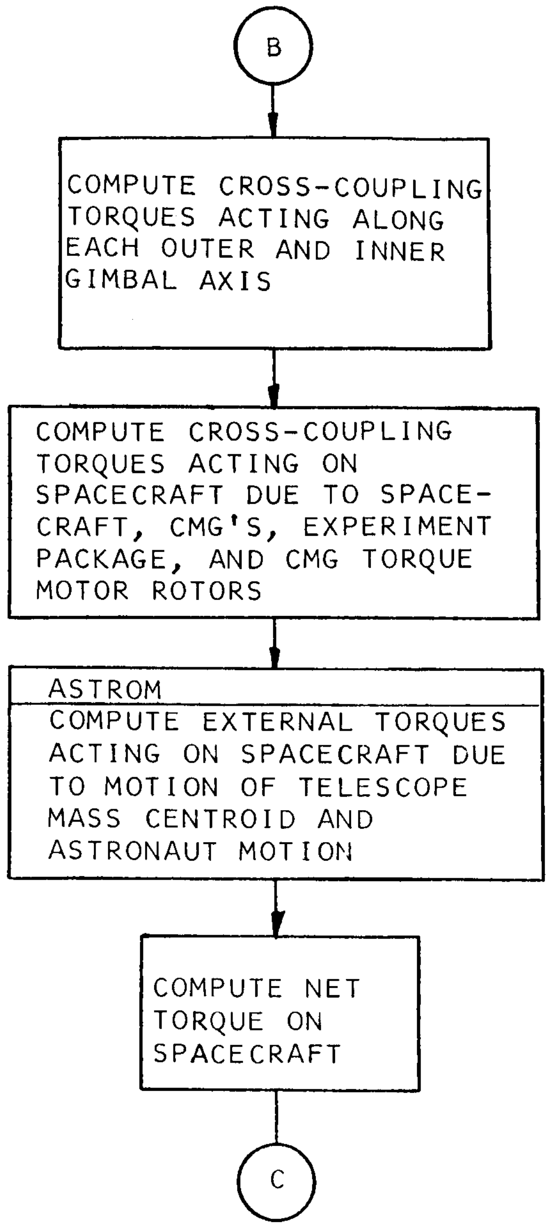
SCATT FLOWCHART



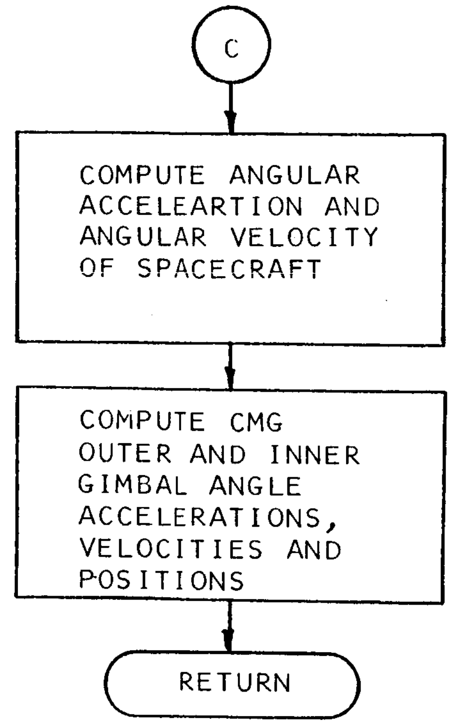
SCATT FLOWCHART (CONTINUED)



SCATT FLOWCHART (CONTINUED)



SCATT FLOWCHART (CONTINUED)



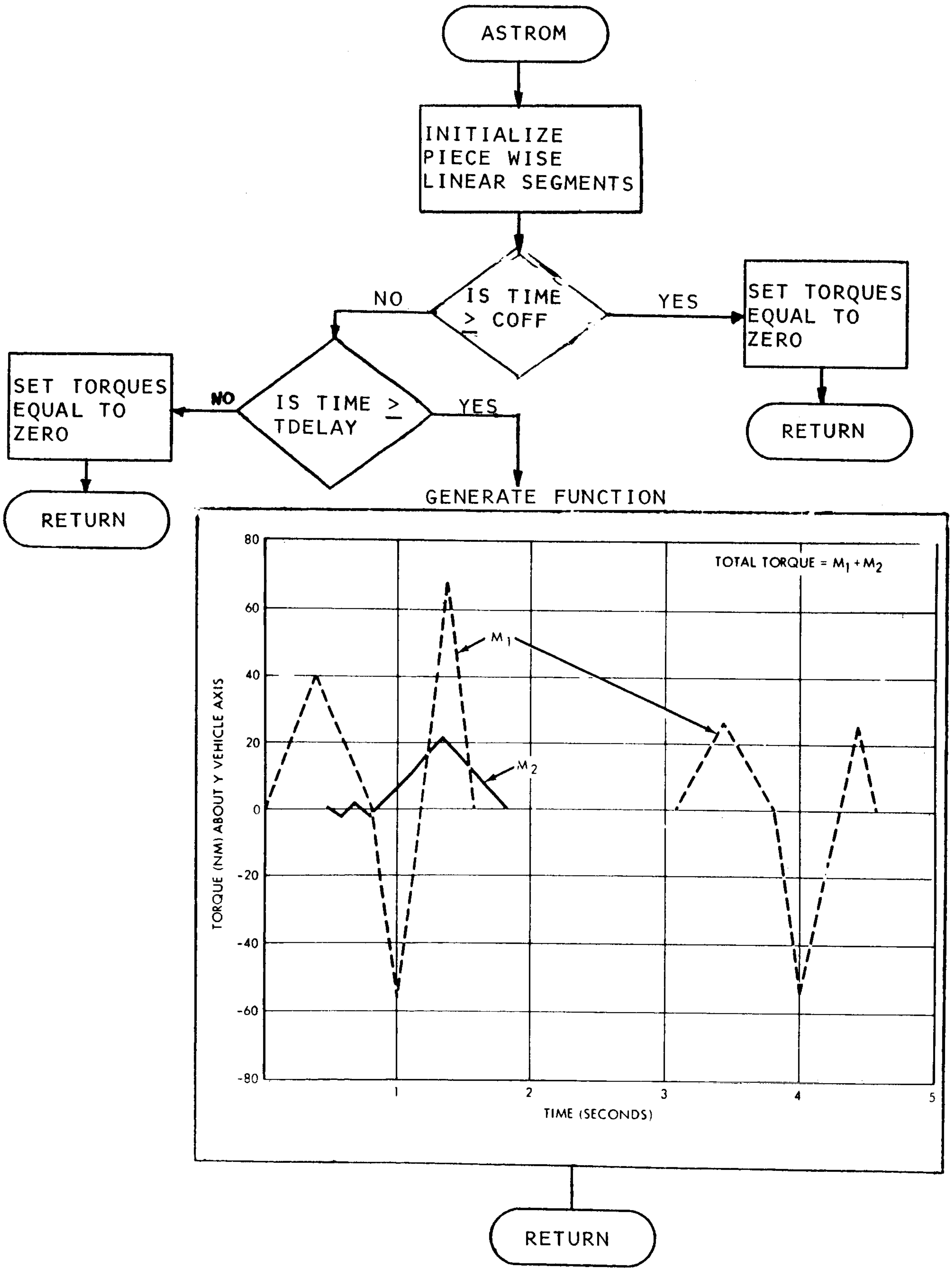
2.3.27 ASTROM Subroutine

Subroutine ASTROM is called once per computation cycle by subroutine SCATT. The ASTROM subroutine is used to generate an external torque profile to simulate astronaut motion. The torque so computed is returned to SCATT where it is added in with other torque terms about the spacecraft mass center.

The torque function generated by ASTROM is shown on the accompanying flowchart. The subroutine may be modified to generate various kinds of external torques. The torque function starts at TDELAY seconds after the simulation starts and will be set zero at COFF seconds into the simulation. To inhibit the astronaut motion torque, COFF should be set zero by data statement in the subroutine itself.

The torque function is generated in terms of several piecewise linear segments reproducing the curve shown.

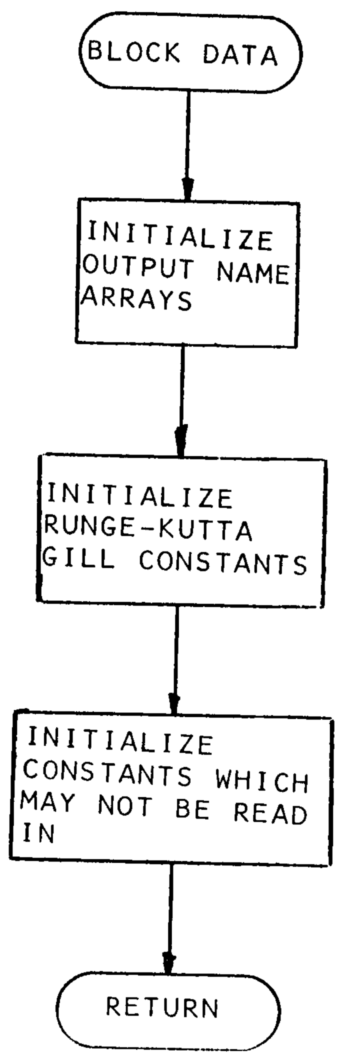
ASTROM FLOWCHART



2.3.28 BLOCK DATA Subroutine

This subroutine is used to initialize constants which are not accessible to the user and which are in common. This routine initializes the constants used in the Runge-Kutta-Gill integration and the arrays which contain the names of output variables. All initialization takes place at compilation time.

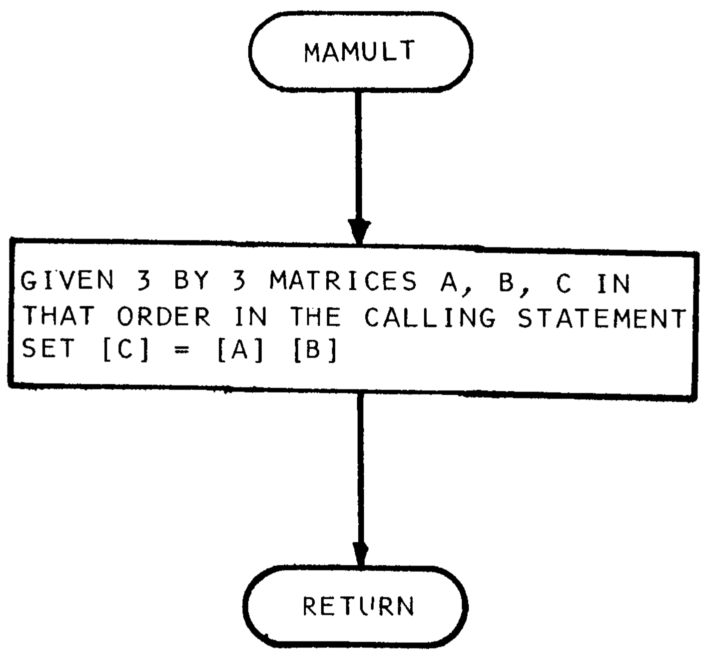
BLOCK DATA FLOWCHART



2.3.29 MAMULT Subroutine

Subroutine MAMULT serves to multiply two 3 by 3 matrices. The names of the matrices to be multiplied and the matrix which will receive the product are specified as the three arguments in the calling statement. Given the three matrices A, B, C in that order in the argument list, MAMULT will set [C] = [A][B]. After performing the matrix product return is made to the calling program.

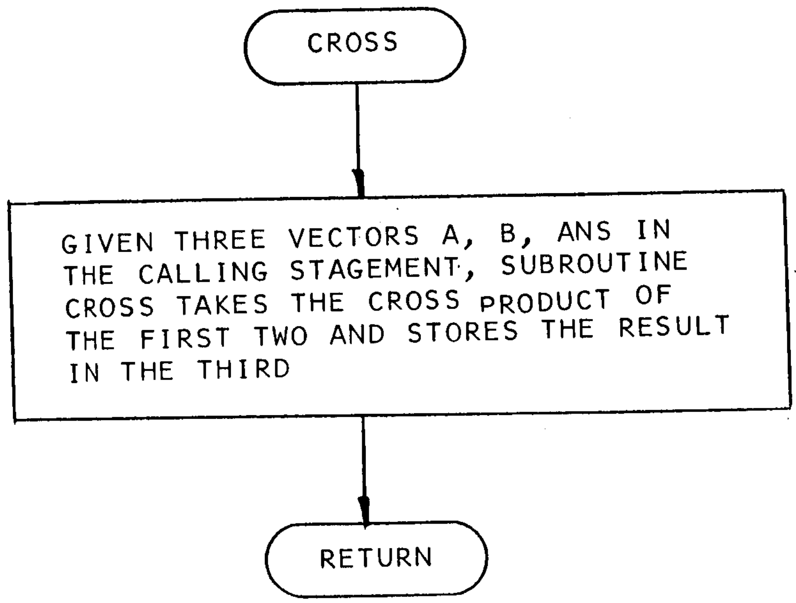
MAMULT FLOWCHART



2.3.30 CROSS Subroutine

Subroutine CROSS is invoked when the cross product of two vectors is required. The two vectors whose cross product is to be obtained and the vector where the product is to be stored are given in the argument list. Given the vectors A, B, ANS, in that order in the argument list, is subroutine sets $\vec{ANS} = \vec{A} \times \vec{B}$. Following execution, return is made to the invoking program.

CROSS FLOWCHART



SECTION 3

PROGRAM TIMING

As illustrated in the preceding sections, timing considerations form an integral part of the program's logic. This paragraph serves to define the fundamental time quantities used in the LASIM program.

MISSION TIME - Mission time is defined to be the length of the time interval over which simulation occurs. The nominal mission time is twenty seconds, but longer simulations may be requested up to 500 seconds.

BASIC TIME STEP - The basic time step is the increment by which time is advanced in the Telescope Control and Spacecraft Attitude Control simulations. Having performed simulation computations for time t, they are next performed for time $t + \Delta t$ where Δt is the basic time step. Having integrated over the interval $t - \Delta t$ to t, integration is performed over the interval t to $t + \Delta t$. The nominal basic time step is ten milliseconds and may be changed by the user.

FINE TIME STEP - The fine time step is the increment by which time is advanced in the Fine Tracking simulation. Having performed fine tracking computations at time t, they are next performed at time $t + \Delta t_f$ where Δt_f is the fine time step. The basic time step will be an integer multiple of the fine time step. The nominal fine time step is 2 milliseconds and may be changed by the user.

ORBIT GENERATING TIME STEP - The orbit generating time step is the increment by which time is advanced in generating an orbit and producing line-of-sight vectors from the orbit to a ground station. Having computed the spacecraft and ground station positions and velocities and the line-of-sight position and velocity vectors at time t, they are next computed at time $t + \Delta t_o$ where Δt_o is the orbiting generating time step. This time step is used only by the orbit generating subprogram and its value will be one-fiftieth of mission time.

ELAPSED TIME - Elapsed time is the time over which the simulation has been run.

PRINT FREQUENCY TIME - Print frequency time is the simulation time that elapses between plot outputs. Plot frequency time must be greater than or equal to one-five hundredth of mission time.

SECTION 4

PROGRAM USAGE

The following paragraphs describe the procedures to be followed in making simulation runs with the LASIM program, and supply information which will allow the user to make use of program features. The LASIM program requires that certain computer hardware and software-system features be available. Thse items are discussed also in the following paragraphs.

4.1 COMPUTER HARDWARE REQUIREMENTS

As mentioned in Paragraph 1.1 of this report, the LASIM program was written for and runs on an IBM 7094 computer with 32,000 word memory, a 1403 printer, and 1402 card reader. Use is made of the SC-4020 plotter for plot output.

The LASIM program requires access to nine tape units in addition to the system input and output units, if use is to be made of all its features. As presently written, specific operations are assigned to specific system tape units. However, the tape unit assignments may be changed in subroutine INIT1 by redesignating the units assigned to the symbolic tape names. Table 4-1 lists the tape units used, the physical designations, the FORTRAN logical unit designations, the functions, and the symbolic program name to which each is assigned, presently. Units A7, B7, A9 and B9 are not assigned by LASIM. If alternate input or output tapes are to be used in place of the system units, these unassigned tape units may be used.

4.2 SYSTEM SOFTWARE REQUIREMENTS

The LASIM program is written in FORTRAN IV, Version 13. The program runs under the IBSYS operating system using the IBJOB processor. All input and output is under control of IOCS. The following system subroutines and library functions are required by the LASIM program.

LSQPF*	DCOS	DLOG	ABS
QUIK3V*	DSIN	DABS	MOD
SMXYV*	DSQRT	DSIGN	ATAN
CLEAN*			

(* - Not standard FORTRAN routines.)

These routines are described in Paragraph 2.1.7.

The system overlay feature is used as discussed in Paragraph 4.6.1.

TABLE 4-1. TAPE UNITS

PHYSICAL UNIT	LOGICAL UNIT	CLASS	FUNCTION	PROGRAM DESIGNATION
A2	5	INPUT	System Input	INPUT
B1	6	OUTPUT	System Output	OUTPUT
A3	1	WORK	Intermediate Input	WKTP1
B3	2	WORK	Intermediate Output	WKTP2
A4	3	WORK	Intermediate Plot Data	WKTP3
B4	4	WORK	Plot Labels	WKTP4
A5	8	WORK	Overlay	--
B5	9	OUTPUT	ORBGEN Output	WKTP9
A6	10	OUTPUT	Pointing Control Data	WKTP10
B6	11	OUTPUT	Restart Data	WKTP11
A8		OUTPUT	SC-4020 Plot Data	WKTP11

NOTE: When a restart job is being run, B6 is input.
When a restart job is to follow, B6 is output.

4.3 COMPUTATION TIME

Computation time on the IBM 7094 computer for a particular simulation run will vary considerably depending upon the amount and type of output, output frequency, time step lengths, and obviously, the mission time duration which will be simulated. The "time ratio" of computation time to mission time in general will vary from approximately 20:1 to 60:1 for the presently programmed time steps, depending upon the amount of output generated. With minimum print output, no plot output, no pointing control tape generation, and using a binary object deck, the "time ratio" should be approximately 20:1.

4.4 USER SUPPLIED INPUT

The following paragraphs describe the format used in preparing input which will be interpreted and processed by the LASIM program. Input is defined herein as either data or control cards which the program will accept and act on to: change the value of a program variable from the nominal in the case of data, or cause the appropriate logic or function to be performed in response to control input.

4.4.1 Control Functions and Control Cards

Table 4-2 contains all the control words recognized and acted on by the LASIM program. If the user desires a function shown in Table 4-2 to be performed, a punched card must be included in the deck setup for each function. Location of control cards in the deck setup is illustrated in Paragraph 4.6.

It is to be noted that all but two control cards are optional. The END card and the /* card <u>must</u> be included in the deck setup. The END card will follow all other data and control cards for a particular run; and the /* card will follow the last END card.

The following card list illustrates the format to be used in preparation of each control card and describes the card. The order in which the cards are listed may be considered the preferred order in which the cards be placed in the deck setup. However, only the END and /* cards <u>must</u> conform to this order.

TABLE 4-2. CONTROL WORDS AND FUNCTIONS

Control Word	Function
COPIES	Optional. Operand specifies number of extra copies of output desired.
DECNTL	Optional. Inhibits execution of Control subprogram.
DEFINE	Optional. Inhibits execution of Fine Tracking System Subprogram.
DESCAT	Optional. Inhibits execution of Spacecraft Attitude Control subprogram.
DETEL	Optional. Inhibits execution of Telescope Control subprogram.
DSPACE	Optional. Causes output to be double spaced.
END	Must be last card of user's input for each mission.
INPUT	Optional. Causes program to use input device specified in operand.
JOB	Optional. Operand becomes heading information for each page of program output.
LINCNT	Optional. Operand specifies number of lines printed per page.
LIST	Optional. Causes listing to be given of all subsequent user supplied input.
OUTPUT	Optional. Causes program to use output device specified in operand.
PAGENO	Optional. Causes operand to be used as first page number on output.
PAHEAD	Optional. Causes an output tape to be generated, on which will be written the necessary parameters to serve as input for the Pointing Control program. The parameters are written every pass through the basic time step loop.

TABLE 4-2. CONTROL WORDS AND FUNCTIONS (CONTINUED)

Control Word	Functions
PLCNTL	Optional. Operand specifies frequency at which variables are plotted.
PLOT	Optional. Causes program to store for plotting, variables whose names appear in operand. Other information in operand becomes coordinate axis labels.
PRCNTL	Optional. Operand specifies frequency at which variables are printed.
PRINT	Optional. Causes program to print values of variables whose names appear in operand.
RESTAF	Optional. Causes storage on tape of all variables needed for subsequent restart run.
RESTAR	Optional. Causes variables and constants stored from previous simulation to be read in and used in a restarted simulation.
RUNOGS	Optional. Causes execution of Orbit Generating subprogram.
TIME	Optional. Operand specifies mission time in seconds.
TIMFIN	Optional. Operand specifies fine time step in milliseconds.
TIMTSC	Optional. Operand specifies basic time step in milliseconds.
TITLE	Optional. Operand becomes title on title page of program output.
USEOGS	Optional. Causes program to use line of sight vector generated by orbit generating subprogram.
/*	Required. Signals termination of job.

JOB CARD - The Job card is an optional first card. The operand, if any, is used as a page heading for each page of the printed output which follows. Only one Job card must be submitted for each mission.

The format of the JOB card is:

```
1             13
+-------------+-------------
JOB    OPERAND
```

LIST CARD - The LIST card causes all succeeding user supplied data cards for the mission to be printed. The List card will also be printed. This card should preceed those data cards which are desired to be printed.

The format of the LIST card is:

```
1
+------------------------------
LIST
```

TITLE CARD - The operand of the TITLE card is used as a title on the title page of the printed output.

The format of the TITLE card is:

```
1             13
+-------------+-----------
TITLE          OPERAND
```

DSPACE CARD - The DSPACE card causes printed output to be double-spaced. If large quantities of output are required, this card should not be used.

The format of DSPACE card is:

```
1
+----------------------------
DSPACE
```

LINCNT CARD - The LINCNT card allows the user to specify the number of lines which will be printed per page. When the number of lines printed on a page equals the number specified in the operand, the page is ejected and printing continues on the new page. The operand for this card will be an integer less than or equal to sixty and must be right adjusted to card column 12.

The format of the LINCNT card is:

```
1                 12
+-----------------+---------------------
LINCNT    OPERAND
```

PAGENO CARD - The operand of this card is used as the page number of the first page of printed output. This card will provide consecutive page numbering for output from a mission and its restart output. The operand for this card will be an integer and must be right adjusted to card column 12.

The format of the PAGENO card is:

```
1                 12
+-----------------+---------------------
PAGENO    OPERAND
```

COPIES CARD - The COPIES card causes extra copies of printed output to be produced. The operand for this card is an integer less than or equal to twenty which specifies the number of extra copies. The operand must be right adjusted to card column 12.

The format of the COPIES card is:

```
1                 12
+-----------------+---------------------
COPIES    OPERAND
```

OUTPUT CARD - The OUTPUT card allows the user to select an alternate output device other than the standard output unit 6. Devices from which he may select an alternate output unit are 12, 13, 14, and 15. The operand will be one of these integers and must be right adjusted to card columns 12.

The format of the OUTPUT card is:

```
1                 12
+-----------------+---------------------
OUTPUT    OPERAND
```

INPUT CARD - The INPUT card allows the user to select an alternate input device other than the standard input unit 5. Devices from which he may select an alternate input unit are 12, 13, 14, and 15. The operand will be one of these integers and must be right adjusted to Column 12.

The format of the INPUT card is:

```
1          12
+----------+--------------
INPUT OPERAND
```

PRINT CARD - The PRINT card allows the user to select variables to be printed. The description and list of names of the variables from which the selection may be made are given in Table 4-4. More than one PRINT card may be used in a mission and up to fifty variables may be selected for printing. The PRINT card may contain more than one variable name; each is the name of a variable selected for printing. The names must be left adjusted to card columns 7, 13, 19, 25, 31, 37, 43, 49, 55, 61, 67, and 73. A variable name on a print card may not begin beyond card column 73.

The format of the PRINT card is:

```
1        7        13       73
+--------+--------+--------+-----------

PRINT    NAME     NAME .... NAME
```

PLOT CARD - The PLOT card allows the user to select variables to be plotted. The description and list of names of the variables from which he may select are given in Table 4-4. More than one PLOT card may be used in a mission and up to twenty pairs of variables may be plotted. The PLOT card contains two variable names; each is the name of a variable selected for plotting. The names must be left adjusted to card columns 7 and 13. The variable whose name begins in card column 7 will be the x-axis variable in the plot. The variable whose name begins in card column 13 will be the y-axis variable. Card column 19 is reserved for a character denoted by c. If c is a blank, both plot axes will have a linear scaling. If c is non-blank, the y-axis will have a log scaling. Card columns 20-39 should contain the x-axis label. Card columns 40-80 should contain the y-axis label.

The format of the PLOT card is:

1	7	13	19	20	39	40	80
PLOT	X-AXIS VARIABLE NAME	Y-AXIS VARIABLE NAME	C	X-AXIS LABEL		Y-AXIS LABEL	

PRCNTL CARD - The PRCNTL card is used to control the frequency at which all variables are printed. The operand of this card will contain an integer greater than zero. The integer specifies the mission time in milliseconds which will elapse between the occurrences of print output. This time is determined by multiplying the basic time step (10 ms, nominally) by the number of program loops desired between plot points. For example, if print is desired every 10th loop, the operand is 100. The operand must be right adjusted to card column 12.

The format of the PRCNTL card is:

1	12
PRCNTL	OPERAND

PLCNTL CARD - The PLCNTL card is used to control the frequency (number of basic time step loops between plot points) at which variables are plotted. The operand of this card will contain an integer greater than or equal to t/250 where t is mission time in milliseconds. The integer operand specifies the simulation time in milliseconds which will elapse between the plot points. The operand must be right adjusted to card column 12.

The format of the PLCNTL card is:

1	12
PLCNTL	OPERAND

TIME CARD - The TIME card allows the user to specify the mission time for a simulation. The operand for this card is an integer less than or equal to five hundred which specifies the length of time in seconds for which the simulation is to occur. The operand must be right adjusted to card column 12.

The format for the TIME card is:

1	12
TIME	OPERAND

TIMTSC CARD - The TIMTSC card allows the user to specify the "basic time step." This is the amount by which time is advanced in the basic time step loop of the program. The operand for this card is an integer representing the basic time step in milliseconds, and must be an integer multiple of the "fine time step." The operand must be right adjusted to card column 12.

The format of the TIMTSC card is:

```
1                 12
+-----------------+--------
TIMTSC      OPERAND
```

TIMFIN Card - The TIMFIN card allows the user to specify the "fine time step." This is the amount by which time is advanced in the fine tracking loop. The operand for this card is an integer representing the fine time step in milliseconds and must be chosen such that the "basic time step" is an integer multiple of the "fine time step." The operand must be right adjusted to card column 12.

The format of the TIMFIN card is:

```
1                 12
+-----------------+----------
TIMFIN      OPERAND
```

RUNTVM CARD - The RUNTVM card causes execution of the Orbit Generating subprogram. It does not cause the line-of-sight vector generated in the Orbit Generating subprogram to be used in the simulation.

The format of the RUNTVM card is:

```
1
+----------------------------
RUNTVM
```

USETVM CARD - The USETVM card causes the line-of-sight vector generated by the Orbit Generating subprogram to be used in the simulation. This card <u>must</u> be accompanied by a RUNTVM card.

The format of the USETVM card is:

```
1
+----------------------------
USETVM
```

DEFIN CARD - This card inhibits execution of the Fine Tracking subprogram.

The format of the DEFINE card is:

```
1
|-------------
DEFINE
```

DETEL - This card inhibits the execution of the Telescope Control subprogram.

The format of the DETEL card is:

```
1
|-------------
DETEL
```

DESCAT CARD - This card inhibits execution of the Space Carft Attitude Control subprogram.

The format of the DESCAT card is:

```
1
|-------------
DESCAT
```

DECNTL CARD - This card inhibits execution of the Control subprogram.

The format of the DECNTL card is:

```
1
|-------------
DECNTL
```

RESTAF CARD - The RESTAF Card signals the program that a restart of the simulation is to follow. This card causes values of variables and constants and the status of program flags and switches to be stored on tape at the termination of the current mission. All words required for subsequent restart of the mission in which this card appears will be stored on logical unit 11, physical unit B6.

The format of the RESTAF card is:

```
1
|--------------------
RESTAF
```

RESTAR CARD - The RESTAR card signals the program that the current run is the restart of a previous simulation. This card causes values of variables and constants and the status of program flags and switches at the termination of a previous mission to be read into their respective program loactions. All words required for the restart of the previous mission are read from logical unit 11, physical unit B6. The storage of user input data follows the reading of the restart tape so that the user may override restart data.

The format of the RESTAR card is:

```
1
+-----------------
RESTAR
```

PAHEAD CARD - The PAHEAD card signals the program that data for the Pointing Control program is to be stored on magnetic tape. All information required as input to the Pointing Control program is written on logical unit 10, physical unit A6 for each basic time step through the simulation.

The format of the PAHEAD card is:

```
1
+----------------------
PAHEAD
```

END CARD - The END card is the last card of the user's source input for a given job. This card is required.

The format of the END card is:

```
1
+----------------
END
```

/* CARD - The /* card is the last card supplied by the user. This card signals the program that all jobs have been processed and causes creation of the final plot tape. This card is required.

The format of the /* card is:

```
1
+----------------------
/*
```

4.4.2 Data Input

This paragraph enumerates the program variables whose value may be changed from the prestored, default value through input data card processing. The data card format to be used is also illustrated.

4.4.2.1 Allowable Input Variables

Table 4-3 contains the variables which will be processed through the input routines of the LASIM program. Only the variables in Table 4-3 may be changed through input data card processing.

4.4.2.2 The Data Card

The data card is used to communicate to the LASIM program the variable and the new value which should be processed by the input routines. More than one variable may be placed upon each data card and variable types may be combined on a single card.

Each data statement must be separated by a comma, and the last data statement on a card must be followed by a comma. Also, column one on each data card must be left blank. Serialization should not be used on a data card since all items on a data card may be scanned.

The following variable forms may be input on data cards:

- Constants: Variable name = constant

  The variable name may be an array element or single program constant. Subscripts must be integer constants.

- Array Name: Array name = set of constants separated by commas

  The number of constants furnished on the card must be equal to the number of elements of the array. The form K* constant may be used to set K array elements to the constant value; where K is an unsigned integer.

o Subscripted variable: Subscripted variable = set of constants separated by commas

A data item of this form results in the set of constant values being placed in consecutive array elements, starting with the element designated by the subscripted variable. The number of constants given cannot exceed the number of array elements included between the given element and the last element in the array. The form K* constant may be used to set K elements equal to a single constant value.

Constants used in the data statements will be of the following types.

o Integer - An integer constant consists of 1 - 11 decimal degits written without a decimal point.

o Real Number - A real constant consists of one of the following:

a. one to nine decimal digits written with a decimal point, but not followed by a decimal exponent.

b. a sequence of decimal digits written with a decimal point, followed by a decimal exponent, which is written as the letter E followed by a signed or unsigned integer constant.

o Double Precision Number - A double precision constant consists of one of the following:

a. ten or more significant decimal digits written with a decimal point, but not followed by a decimal exponent.

b. a sequence of decimal digits written with or without a decimal point, followed by a decimal exponent, which is written as the letter D followed by a signed or unsigned integer constant.

These types of constants may be associated with integer, real or double precision data words and are converted in accordance with the type of data word. Blanks must not be embedded in a constant or repeated constant field, but may be used freely elsewhere on a data card.

The following illustrates the format used in preparing the data card in the form of two examples.

First Data Card
```
1
b ALPH = 3*.5D-8, LAD = 36, LAM = 40,
```

Last Data Card
```
1
b AO = 3.007D-6, BETA (2) = .2D-9,
```

The examples illustrate that the first character must be blank. If this data is input to the LASIM program, the double precision number .5 x 10^{-8} is placed in ALPH (1), ALPH (2), and ALPH (3). Since ALPH is an array name not followed by subscript, the entire array is filled with the succeeding constants. The integers 36 and 40 are placed in LAD and LAM respectively. The constant 3.007 x 10^{-6} is stored in AO and .6 x 10^{-9} is stored in BETA (2).

Continuation of a data statement may not be made onto a succeeding card, since the last character on each card must be a comma indicating the end of a data statement.

TABLE 4-3. INPUT DATA WORDS AND DESCRIPTION

ITEM	MATH SYMBOL	FORM	TYPE	DESCRIPTION	DEFAULT VALUE	UNITS
A[1]	A_1	C	DP	Principal moments of inertia of telescope inner gimbal about x_{11} axis.	16.8	$kg\text{-}m^2$
ALPH(1)	α_1	C	DP	Outer gimbal of CMG #1	0.785398	radians
ALPH(2)	α_2	C	DP	Outer gimbal of CMG #2	0.785398	radians
ALPH(3)	α_3	C	DP	Outer gimbal of CMG #3	0.785398	radians
ALPHX	α_x	C	DP	Pitch telescope offset angle	0	radians
ALPHY	α_y	C	DP	Yaw telescope offset angle	0	radians
AO	A_o	C	DP	Principal moments of inertia of telescope outer gimbal about X_{10} axis	32.5	$kg\text{-}m^2$
AT	A_T	C	DP	Principal moments of inertia of telescope experiment package about Y_T	2111.0	$kg\text{-}m^2$
B[1]	B_1	C	DP	Principal moments of inertia of telescope inner gimbal about y_{11} axis.	3.5	$kg\text{-}m^2$
BETA(1)	β_1	C	DP	Inner gimbal of CMG #1	0	radians
BETA(2)	β_2	C	DP	Inner gimbal of CMG #2	0	radians
BETA(3)	β_3	C	DP	Inner gimbal of CMG #3	0	radians

Table 4-3. Input Data Words and Description (Continued)

ITEM	MATH SYMBOL	FORM	TYPE	DESCRIPTION	DEFAULT VALUE	UNITS
BO	B_o	C	DP	Principal moments of inertia of telescope outer gimbal about y_{10} axis.	130.2	$kg\text{-}m^2$
BT	B_T	C	DP	Principal moments of inertia of telescope experiment package about Y_T	2111.0	$kg\text{-}m^2$
C1	C_1	C	DP	Principal moments of inertia of telescope inner gimbal about z_{11} axis	15.0	$kg\text{-}m^2$
CMG1Z	G_{1Z}	C	DP	Vehicle control law position loop gain for Z_B axis	1.68×10^6	n-m/rad
CMGPLM	CMGPLM	C	DP	Vehicle position torque command limit	1500	nm
CMGCLM	CMGCLM	C	DP	Vehicle total torque command limit	408	nm
CO	C_o	C	DP	Principal moments of inertia of telescope outer gimbal about z_{10} axis	119.3	$kg\text{-}m^2$
CT	C_T	C	DP	Principal moments of inertia of telescope experiment package about z_T	954	$kg\text{-}m^2$
DA	A	C	DP	Servo amp open loop gain	10^6	volt/volt
DA3	A_3	C	DP	Transfer lens motor scale factor.	2.67×10^4	dynes/volt
DB	B	C	DP	Transfer lens motor time constant.	$\frac{1}{2} \pi$ (780)	sec

Table 4-3. Input Data Words and Description (Continued)

ITEM	MATH SYMBOL	FORM	TYPE	DESCRIPTION	DEFAULT VALUE	UNITS
DKO	KO	C	DP	Voltage divider gain fine tracking system	4.1252×10^{-6}	volt/volt
DK1	K_1	C	DP	Transfer lens dynamics scale factor	1/250	cm/dyne
DK2	K_2	C	DP	Lens dynamics break frequency squared	141.4	r^2/sec^2
DK3	K_3	C	DP	Velocity sensor gain	0.0167	volts/cm/sec
DK4	K_4	C	DP	Velocity sensor break frequency	2π (120)	rad/sec
DT1	T_1	C	DP	Lead-Lag Network time constant	$1/0.932\pi$	sec
DT2	T_2	C	DP	Lead-Lag Network time constant	$1/14\pi$	sec
DT4	T_4	C	DP	Servo amp feedback network time constants	$1/2\pi(20.1)$	sec
DT6	T_6	C	DP	Intermediate variable	AC + T4	sec
F	K_o	C	DP	Fine optical system gain	7.92×10^{-10}	volts/photon/sec
GCMG1	G_1	C	DP	Vehicle control law position loop gain for $Y_B X_{B1}$ axes	1.3436×10^{T}	n-m/rad
H1	H_1	C	DP	Angular momentum for CMG #1 rotor	2720	n-m-s
H2	H_2	C	DP	Angular momentum of CMG #2 rotor	2720	n-m-s
H3	H_3	C	DP	Angular momentum of CMG #3 rotor	2720	n-m-s

Table 4-3. Input Data Words and Description (Continued)

ITEM	MATH SYMBOL	FORM	TYPE	DESCRIPTION	DEFAULT VALUE	UNITS
IXXP	I_{XX}	C	DP	Moment of inertia of LM/CSM plus ATM rack (main body) about X_B	341473	$kg\text{-}m^2$
IXYP	I_{XY}	C	DP	Products of inertia of main body	0	$kg\text{-}m^2$
IXZP	I_{XZ}	C	DP	Products of inertia of main body	0	$kg\text{-}m^2$
IYYP	I_{YY}	C	DP	Moment of inertia of main body about Y_B	341473	$kg\text{-}m^2$
IYZP	I_{YZ}	C	DP	Products of inertia of main body	0	$kg\text{-}m^2$
IZZP	I_{ZZ}	C	DP	Moment of inertia of main body about Z_B	42730	$kg\text{-}m^2$
LAD	λ_g	C	I	Ground station latitude degrees	32	degrees
LAM	λ_g	C	I	Ground Station latitude minutes	12	minutes
LAS	λ_g	C	I	Ground station latitude seconds	49	seconds
LOD	ϕ_g	C	I	Ground station longitude degrees	- 104	degrees
LOM	ϕ_g	C	I	Ground station longitude minutes	54	minutes
LOS	ϕ_g	C	I	Ground station longitude seconds	49	seconds

Table 4-3. Input Data Words and Description (Continued)

ITEM	MATH SYMBOL	FORM	TYPE	DESCRIPTION	DEFAULT VALUE	UNITS
OBO	λ_s	C	R	Insertion latitude + ° north − ° south	0	degrees
OSO	ω	C	R	Argument of Perigee	0	degrees
OSI	i	C	R	Inclination of orbit	28.3	degrees
OSE	e	C	R	Eccentricity	0.05	
OBT	ϕ_s	C	R	Insertion longitude	- 104	degrees
OPER	τ	C	R	Period of the orbit	86160.244	seconds
OSA	a	C	R	Semi-major axis		feet
ORP	R_p	C	R	Perigee altitude		nautical miles
ORA	R_a	C	R	Apogee altitude		nautical miles
P	P	C	DP	Spacecraft angular velocity about X_B	0	rad/sec
PSI1	ψ_1	C	DP	Telescope outer gimbal	0	radians
PSI2	ψ_2	C	DP	Telescope inner gimbal	0	radians
Q	Q	C	DP	Spacecraft angular velocity about Y_B	0	rad/sec
R	R	C	DP	Spacecraft angular velocity about Z_B	0	rad/sec
TCMG1	1	C	DP	Vehicle control law time constant	0.225 s	sec
TCMG2	2	C	DP	Vehicle control law lag term time constant	0.0053	sec

Table 4-3. Input Data Words and Description (Continued)

ITEM	MATH SYMBOL	FORM	TYPE	DESCRIPTION	DEFAULT VALUE	UNITS
TLX	t_x	C	DP	Transfer lens' x position coordinate	0	cm
TLY	t_y	C	DP	Transfer lens' y position coordinate	0	cm

LEGEND:

FORM	TYPE
C = constant	DP = double precision
V = variable	R = real
A = variable array	I = integer

4.5 Program Output

Program output will be in the form of:

- o Printed Output
- o Plot Output
- o Magnetic Tape Output

The following paragraphs describe each of these separate classes of output.

4.5.1 Printed Output

Printed output falls into the following categories.

AUTOMATIC PRINTED OUTPUT - Certain important program variables are printed without a request from the user. Further, the user cannot inhibit printing of these variables. The frequency at which they are printed is under user control as discussed in Paragraph 4.4. Table 4-4 contains the automatically printed variables.

SELECTED VARIABLES - The user may request that certain program variables be printed in addition to those over which he has no control. The second part of Table 4-4 lists the variables from which the user may select. These variables will be printed at the frequency selected by the user for printed program output. The necessary control word input to cause selected variables to be printed is discussed in Paragraph 4.4

OPTIONAL PRELIST - The user may, by appropriate control card input, cause the data card images to be printed. See Paragraph 4.4 for the action required.

DIAGNOSTIC MESSAGES - In addition to the optional Prelist, subroutine PRIN provides diagnostics during the processing of the input. Preceding each diagnostic is the sequence number assigned to each input record by the program. The sequence number and diagnostic appear to the right of the printed input line for easy recognition. Some diagnostics are warnings, others will inhibit the simulation. Other error messages are printed in subroutine TANDR when inconsistencies are found in the input. All diagnostic and error messages are listed in Table 4-5. Those without an associated IER Flag are printed in subroutine TANDR. The rest are printed in subroutine PRIN.

TABLE 4-4. PROGRAM OUTPUT VARIABLES

Program Name	Description	Units
	- AUTOMATICALLY PRINTED VARIABLES -	
DDPSI1	Telescope outer gimbal angle acceleration.	rad/sec^2
DDPSI2	Telescope inner gimbal angle acceleration.	rad/sec^2
DP	X_B component of angular acceleration of main body.	rad/sec^2
DPSI1	Telescope outer gimbal angle rate.	rad/sec
DPSI2	Telescope inner gimbal angle rate.	rad/sec
DQ	Y_B component of angular acceleration of main body.	rad/sec^2
DR	Z_B component of angular acceleration of main body.	rad/sec^2
EEX	Transfer lens servo error voltage in X channel.	volts
EEY	Transfer lens servo error voltage in Y channel.	volts
EPX	Total tracking error in X direction.	arc-sec
EPY	Total tracking error in Y direction	arc-sec
P	X_B component of angular velocity $\underline{\omega}$.	rad/sec
PSI1	Telescope outer gimbal angle.	rad
PSI2	Telescope inner gimbal angle.	rad
Q	Y_B component of angular velocity $\underline{\omega}$.	rad/sec
R	Z_B component of angular velocity $\underline{\omega}$.	rad/sec
SD1 SD2 SD3	X Y Z component of line-of-sight vector in [T] frame.	
THETAZ	Angle between telescope longitudinal axis and line-of-sight.	arc seconds
TTX	Transfer lens X position coordinate.	cm
TTY	Transfer lens Y position coordinate.	cm
X	Elapsed time.	seconds
	- OPTIONAL PRINTED VARIABLES -	
A11 A12 A13 A21 A22 A23 A31 A32 A33 A41 A42 A43	Cross coupling torque acting on spacecraft due to CMG's and Experiment Package.	n-m

TABLE 4-4. PROGRAM OUTPUT VARIABLES (CONTINUED)

Program Name	Description	Units
Optional Printed Variables (Continued)		
ALPH	Outer gimbal angle of CMG's 1, 2 and 3.	radians
ANGEX	Telescope X position error signal in fine mode.	volts
ANGEY	Telescope Y position error signal in fine mode.	volts
AX AY AZ	X, Y, Z components, polynomial coefficients for line of sight.	
B	Alternate usage for T2I matrix.	
B4 B5 B6 B7 B8 B9	Cross-coupling torques acting along each CMG outer and inner gimbal axis.	n-m
B2I	Matrix which relates the actual initial telescope frame to the nominal initial telescope frame.	
BETA	Inner gimbal angle of CMG's 1, 2 and 3.	rad
CPSI1	Trignometric cosine of ψ_1.	
CPSI2	Trignometric cosine of ψ_2.	
DALP1C DALP2C DALP3C	Commanded CMG outer gimbal angle rates (primed-before limiting, unprimed-after limiting).	rad/sec
DALP1E DALP2E DALP3E	CMG outer gimbal angle rate errors.	volt
DALPH	Outer gimbal angle rate of CMG's 1, 2 and 3.	rad/sec
DBET1C DBET2C DBET3C	Commanded CMG inner gimbal angle rates (Primed-before limiting, unprimed-after limiting)	rad/sec
DBET1E DBET2E DBET3E	CMG inner gimbal angle rate errors.	volts
DBETA	Inner gimbal angle rate of CMG's 1, 2 and 3.	rad/sec
DDALPH	Outer gimbal angular acceleration of CMG's 1, 2 and 3.	rad/sec^2
DDBETA	Inner gimbal angular acceleration of CMG's 1, 2 and 3.	rad/sec^2
DALPHX	Component of inertial angular rate of telescope about X_T.	rad/sec
DALPHY	Component of inertial angular rate of telescope about Y_T.	rad/sec

TABLE 4-4. PROGRAM OUTPUT VARIABLES (CONTINUED)

Program Name	Description	Units
Optional Printed Variables (Continued)		
DLX	Transfer lens rate - X component.	cm/sec
DLY	Transfer lens rate - Y component.	cm/sec
INT	Total light energy.	photons/sec
LMX LMY LMZ	X Y component in inertial system of Z $\underline{L}_M$.	meters
LMAG	Magnitude of line-of-sight vector.	meters
MA1 MA2 MA3	Torque produced by CMG's 1, 2 and 3 outer gimbal torques after reflection through gearing.	n-m
MB1 MB2 MB3	Torque produced by CMG's 1, 2 and 3 inner gimbal torques after reflection through gearing.	n-m
MT1	Output of telescope outer gimbal torque motor after limiting.	n-m
MT2	Output of telescope inner gimbal torque motor after limiting.	n-m
POSEX POSEY	Position error commands from an ideal coarse or fine optical sensor.	volt
POSXLI POSYLI	Position error commands from actual coarse of fine optical sensor.	volt
PX	Image Center X-coordinate in f/70 plane.	cm
PY	Image Center Y-coordinate in f/70 plane.	cm
RATEXI	Component of measured inertial angular rate of telescope about X_T.	rad/sec
RATEYI	Component of measured inertial angular rate of telescope about Y_T.	rad/sec
RATIOX	Energy fraction.	
RATIOY	Ener y fraction	
SB1 SB2 SB3	Cross-coupling torques acting on space-craft due to spacecraft CMG's, telescope, and CMG torque motor ratios.	n/m
SB10 SB11	Cross-coupling torques acting along telescope outer and inner gimbal axes.	n/m
SPSI1	Trignometric sine of ψ_1.	
SPSI2	Trignometric sine of ψ_2.	
T2B	Telescope to Body transformation matrix.	
T2I	Telescope to Inertial transformation matrix.	
TMA1 TMA2 TMA3	Intermediate parameters in CMG velocity loop.	volt
TMB1 TMB2 TMB3	Intermediate parameters in CMG velocity loop.	volt

TABLE 4-4. PROGRAM OUTPUT VARIABLES (CONTINUED)

Program Name	Description	Units
	Optional Printed Variables (Continued)	
WX4 WY4 WZ4	X Y component of telescope angular velocity in telescope frame. Z	rad/sec
WX4P WY4P WZ4P	Component of $\underline{\omega}_4'$ relative to telescope outer gimbal coordinates.	rad/sec

TABLE 4-5. PROGRAM DIAGNOSTIC MESSAGES

ERROR MESSAGE	CAUSE	PROGRAM ACTION	ERROR MESSAGE POINTER - IER-
'Extra Job Card Deleted	The user has supplied more than one Job Control Card	Extra job card is ignored. First Job Card is used.	IER=2
'Blank Operand In Copy Control Card'	Number of output copies not specified in Copy Control Card.	Only one output copy will be produced.	IER=3
'Blank Operand in Print Control Card'	Print frequency not specified in PRCNTL Control Card	The program will formulate a reasonable output frequency based on number of passes through the simulation.	IER=4
'Blank operand in Plot Control Card'	Plot frequency not specified in PLCNTL Control Card	The program will formulate a reasonable output frequency based on number of passes through the simulation.	IER=5
'Blank Operand for Simulation Time'	Mission time not specified in Time Control Card.	Mission time will assume default value.	IER=6
'Blank Operand for Simulation Time Step'	Simulation time step not specified in TIMTSC or TIMFIN Control Card.	Time step will assume	IER=7
'Invlaid output device request ignored'	The user has requested for output a device which is not available.	Request will be ignored.	IER=8

TABLE 4-5. PROGRAM DIAGNOSTIC MESSAGES (CONTINUED)

ERROR MESSAGE	CAUSE	PROGRAM ACTION	ERROR MESSAGES POINTER - IER-
'Invalid input device request ignored'	The user has requrested for output a device which is not available	Request will be ignored.	IER=9
'Requested lines per page exceeds maximum'	The user has requested more lines per page than are available.	The program will use maximum allowable lines per page.	IER=10
'Invalid Card - Job terminated'	A user supplied card has been received for which no remedial action could taken.	Program will be terminated following the processing of all user supplied input.	IER=11
'Invalid x - Variable plot request ignored'	The user has requested unavailable variable for plot.	Program will ignore plot request.	IER=12
'Invalid y - Variable plot request ignored'	The user has requested unavailable variable for plot	Program will ignore plot request	IER=13
'Invalid print request ignored'	The user has requested print output for unavailable variable.	Program will ignore print request.	IER=14
'Output Request Exceeds Maximum-Ignored'	The user has requested more than fifty variables for printing or more than twenty plots.	This and additional output requests will be ignored.	IER=15

TABLE 4-5. PROGRAM DIAGNOSTIC MESSAGES (CONTINUED)

ERROR MESSAGE	CAUSE	PROGRAM ACTION	ERROR MESSAGES POINTER - IER-
'Operand Not Right adjusted - Card Ignored'	The user has not right adjusted the numeric operand in a control card.	Program will ignore Control card.	IER=16
'Operand Not Valid-Card Ignored'	The user has specified an invalid Control Card operand.	Program will ignore Control Card.	IER=17
'Input and output Must Not Be The Same'	The user has requested that the same device be used for both input and output.	Output will be assigned to logical unit 6.	
'Mission Time Request Too Large - Now set at 500 seconds'	The user has requested a mission time that exceeds 500 seconds.	Mission time will be set at 500 seconds.	
'Basic Time Step Must be a Multiple of Fine Time Step'	Basic time step is not a multiple of the Fine Time Step.	Basic time step will be initialized by program.	
'Plot frequency request too fast - Request ignored'	Request has been received which produces excessive plot output.	Request will be ignored. Default plot frequency will be assumed.	

ORBIT GENERATOR OUTPUT - If the user requests execution of the Orbit Generating Subprogram, output related to the orbit and line of sight vectors will be printed. A description of the items printed accompanies each item in the output. This output is not under control of the user except that he may or maynot invoke the orbit generating subprogram.

4.5.2 Plot Output

Plots of certain program variables may be requested by the user. The list of variables from which the user may select includes all the variables listed in Table 4-4. The procedures to be followed in requesting plots are explained in Paragraph 4.4.

4.5.3 Magnetic Tape Output

Three special purpose tapes are generated by the LASIM program in addition to the necessary tapes for printed output, plots, and multiple copies of printout. The logical units on which these tapes appear and the program designation of these tapes is given in Table 4-1.

4.5.3.1 Pointing Control Tape

The first of these tapes is generated to serve as the input tape for the Pointing Control program. Generation of this tape is optional (See Paragraph 4.4). If the pointing functions of the LCSE are to be simulated for a particular mission, the Pointing Control tape must be generated for use by the Pointing Control program.

The variables shown in Table 4-6 are written in the indicated order on the Pointing Control Tape very basic time step through the LASIM program in binary.

TABLE 4-6. POINTING CONTROL TAPE CONTENTS

Variable	Description
LMX	Line-of-sight x component in inertial coordinates.
LMY	Line-of-sight y component in inertial coordinates.
LMZ	Line-of-sight z component in inertial coordinates.
VLMX	Line-of-sight velocity x coordinate in inertial coordinates.
VLMY	Line-of-sight velocity y coordinate in inertial coordinates.
VLMZ	Line-of-sight velocity z coordinate in inertial coordinates.
T2I	Telescope-to-inertial transformation matrix (9 elements)
TLX	Transfer lens x position coordinate.
TLY	Transfer lens y position coordinate.

The Pointing Control tape must be saved from a LASIM program run and used as an input tape for the Pointing Control program. The use to which this tape is put in the Pointing Control program is discussed in Paragraph 6 of this report.

4.5.3.2 Restart Tape

If a "restart job" is requested, a Restart Tape is generated at the end of the LASIM program simulation run which is to be "restarted." This tape is used as input in the "restarted" run. The variables and their values which are written on this tape are numerous and will not be listed here. All the program variables contained in all program "common blocks" except BLOCK 3 will be written on the Restart Tape. Reference to the LASIM Program Listing will identify the variables which are written on the Restart Tape.

4.5.3. Orbit-Generator Output Tape

When the Orbit Generating subprogram is used, a byproduct of this routine is the generation of a tape containing line-of-sight and related parameters. The output listed below is stored on the tape in binary in fifty records, each record consisting of twenty one words. A record is written each pass through the orbit generating routine of which there are fifty. The following list defines the content of each word in a record. No use is presently made of this tape, however, it was felt some future application could possibly use this and the tape generation left in the program.

ORBIT GENERATOR OUTPUT TAPE WORD LIST

Word No.	Content
1	Ground Station X coordinate in inertial coordinates.
2	Ground Station Y coordinate in inertial coordinates.
3	Ground Station Z coordinate in inertial coordinates.
4	Ground Station Velocity X coordinate in inertial coordinates.
5	Ground Station Velocity Y coordinate in inertial coordinates.
6	Ground Station Velocity Z coordinate in inertial coordinates.
7	Spacecraft Position X coordinate in inertial coordinates.
8	Spacecraft Position Y coordinate in inertial coordinates.
9	Spacecraft Position Z coordinate in inertial coordinates.
10	Spacecraft Velocity X coordinate in inertial coordinates.
11	Spacecraft Velocity Y coordinate in inertial coordinates.
12	Spacecraft Velocity Z coordinate in inertial coordinates.
13	Spacecraft Acceleration X coordinate in inertial coordinates.
14	Spacecraft Acceleration Y coordinate in inertial coordinates.
15	Spacecraft Acceleration Z coordinate in inertial coordinates.
16	Line of sight X coordinate in inertial coordinates.
17	Line of sight Y coordinate in inertial coordinates.
18	Line of sight Z coordinate in inertial coordinates.

Orbit Generator Output Tape Word List (Continued)

Word No.	Content
19	Line of sight velocity X coordinate in inertial coordinates.
20	Line of sight velocity Y coordinate in inertial coordinates.
21	Line of sight velocity Z coordinate in inertial coordinates.

4.6 DECK SETUP

This section describes procedures for arranging the program deck, Control cards, and Data cards. Figure 4-1 illustrates graphically, the deck setup. The cards indicated at the front of the entire program decks are standard control cards used by the IBSYS system. The source or object program deck is separated into four segments or links as indicated in Fugure 4-1. The subroutine decks comprising Link 1, Link 2 and Link 3 must be preceded by a $ORIGIN card. This organization is required for overlay as described in Paragraph 4.6.1.

The complete LASIM program deck, either the source or object deck, will have a $DATA card as its last card. The user will place his Control cards and Data cards behind the $DATA card. These will be grouped by mission and the last card for each group will be the END control card. Following the last END control card, the user will insert a /* control card. The last card in the deck, which will be placed behind the /* card is the End of File card (7 and 8 punch in Column 1).

Figure 4-2 shows a sample instruction card which must accompany a completed deck setup in order to be run on the MSFC Computation Laboratory 7094 facility. In the sample illustrated, ten plots have been requested, program work tapes designated, an Orbit Generator Output Tape created (B5) and a Pointing Control Tape created (A6) and saved. For tape assignments reference is made to Table 4-1.

```
                              7 8
                              /*
                              END
                              CONTROL AND DATA CARDS
                              FOR LAST MISSION

                              END
                              CONTROL AND DATA CARDS
                              FOR FIRST MISSION

                              $DATA
                              LINK3  PROGRAM DECK
                              SOURCE OR OBJECT

$IBFTC cards→                 $IBFTC   DECK37 LIST,REF,NODECK,M94
replaced by                   $ORIGIN
$IBLOR cards                  LINK2  PROGRAM DECKS
when using                    SOURCE OR OBJECT
object decks
(Same as     →                $IBFTC   DECK20  LIST,REF,NODECK,M94
 above)                       $ORIGIN
                              LINK1 PROGRAM DECKS
                              SOURCE OR OBJECT

(Same as     →                $IBFTC   DECK4  LIST,REF,NODECK,M94
 above)                       $ORIGIN
                              LINKO PROGRAM DECKS
                              SOURCE OR OBJECT

(Same as     →                $IBFTC DECK1  LIST,REF,NODECK,M94
 above)                       $ETC -UNIT08-,-UNIT09-,-UNIT10-,-UNIT11-,OPNCT=10,BUFCT=10
                              $GROUP -UNIT01-,-UNIT02-,-UNIT03-,-UNIT04-,-UNIT05-,-UNIT06-,
                              $POOL -UNIT01-,BLOCK35,BUFCT-10
                              $FILE -UNIT15-NONE
                              $FILE -UNIT14-NONE
                              $FILE -UNIT13-NONE
                              $FILE -UNIT12-NONE
                              $FILE -UNIT07-NONE
                              $IBJOB  GO,LOGIC,MAP,FILES
                              $EXECUTE  IBJOB
                              $JOB CARD
```

FIGURE 4-1. LASIM PROGRAM DECK SETUP

7094-________ INSTRUCTIONS

NAME: John Doe		OP CODE: 2	STACK #
BIN # 2	LOC: BLDG 4471	JOB: # 444888	

IF EXCEEDS MAX: ☒ STR ☐ STZ ☐ DMP ☐ RETSY

☐ IBSYS ☐ SPOOK ☐ OTHER

☒ COMPL /ASSMBL ☒ EXECUTE ☐ PUNCH (BCD BIN)

☒ 4 FTRN ☐ 2 FTRN ☐ APT ☐ PERT

☐ MAP ☐ FAP ☐ SCAT ☐ OTHER

FAST TAPES: A B C D

INPUT TAPES			WORK
LOGIC	REEL NO.	DEN	LOGIC
			A3
			A4
			B3
			B4
			A5

LINES OF OUTPUT (1000'S)

☐ 0-5 ☐ 5-15 ☒ 15-30 ☐ OVER

MAXIMUM TIME: HOURS 0 MINUTES 20

PROGRAMMER COMMENTS:

NUMBER OF CASES ________

OVER: ________

☐ SEE ON-LINE
☐ SEE TECHNIQUE
☐ MAX EXCEEDED
☐ RETURN TO SYS
☐ LINE MAX

OPERATOR COMMENTS:

OPER INIT: ________
OVER: ________

OUTPUT TAPES ONLY						4020
REEL NO.	LOGIC	DEN.	UNIT	NO OF CPYS	SAVE	TAPE
	B-1	8				
	B5					
	A6				X	
	B6					
	A8	5				118

NO FILES	NO FRAMES	COPIES		DENSITY		COPY-FLO		KALVAR
		P	F	5	8	P	F	
1	10	1	1	X				

MSFC - Form 533 (Rev February 1966)

FIGURE 4-2. USER'S INSTRUCTION CARD

4.6.1 Memory Overlay

The overlay feature of the IBYSIS system is used in order to load the entire LASIM program into computer memory. A simple overly structure is used for the LASIM program, consisting of four links. Link 0 remains in core at all times. Link 1 is loaded and later replaced by Link 2 which is replaced by Link 3. Table 4-7 lists the LASIM subroutines and functions contained in the four links in the order in which they are loaded. System routines are not shown. The program listing illustrates the complete memory map. Table 4-7 also illustrates the deck numbers associated with each LASIM program deck as they appear on $IBFTC cards and in the listing.

TABLE 4-7. PROGRAM DECK SEQUENCE

- LINK0 -

Deck Number	Subroutine	Sequence In Link 0
1	EXEC	1
2	MAMULT	2
3	CROSS	3

- LINK1 -

Deck Number	Subroutine	Sequence In Link 1
	INIT1	
4	INITR	1
	INIT2	
5	CHKCRD	2
6	PROCON	3
7	PRODAT	4
8	PRIN	5
9	TANDR	6
10	REST	7
11	ORBGEN	8
12	PVINO	9
13	DFLCW	10
14	DERIV	11
15	OPUT	12
16	ANGLES	13
17	ACOS	14
18	ASIN	15
19	ATAN2	16

TABLE 4-7. PROGRAM DECK SEQUENCE (CONTINUED)

- LINK2 -

Deck Number	Subroutine	Sequence In Link 2
20	FINE	1
21	CALC2	2
22	CALC3	3
23	XYCURVE	4
24	RNG	5
25	INTENS	6
26	TELCON	7
27	CONTRL CONTL2	8
28	DIRCOS	9
29	SCATT	10
30	ASTROM	11
31	F1	12
32	F2	13
33	F3	14
34	OUTPLT OUTPL	15
35	OUTPRT	16
36	BLOCK DATA	17

- LINK3 -

Deck Number	Subroutine	Sequence In Link 3
37	OUTPLF	1

A $ORIGIN card <u>must</u> precede the first subroutine deck in Links 1, 2 and 3 but <u>not</u> Link 0. The deck set illustrated in Figure 4-1 shows this. The $ORIGIN cards up are required to actuate the system overlay feature which is required.

SECTION 5

LASIM PROGRAM DICTIONARY

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
AA	A_a	2.1	D	11.93		INIT1	Principal moment of inertia of CMG outer gimbal (k_g - m^2)
AB	A_b	2.1	D	1.50		INIT1	Principal moment of inertia of CMG inner gimbal (k_g - m^2)
AG	A_g	2.1	D	3.30		INIT1	Principal moment of inertia of CMG gyro rotor (k_g - m^2)
AI	A_I	2.1	D	16.8		INIT1	Principal moment of inertia of telescope inner gimbal (kg - m^2)
ALOSX	L_x	2.7	R		0	ORBGEN	Line-of-sight x-coordinate in inertial frame
ALOSY	L_y	2.7	R		0	ORBGEN	Line-of-sight y-coordinate in inertial frame
ALOSZ	L_z	2.7	R		0	ORBGEN	Line-of-sight z-coordinate in inertial frame
ALPHX	α_x	2.6	D	0		INIT2	Telescope pitch offset angle (rad)
ALPHY	α_y	2.6	D	0		INIT2	Telescope yaw offset angle (rad)
ANGEX	$L_x(\frac{K_1}{L_1})$	2.4.2.5	D		0	TELCON	Telescope x position error signal in fine mode (volts)
ANGEY	$L_y(\frac{K_1}{L_1})$	2.4.2.5	D		0	TELCON	Telescope y position error signal in fine mode (volts)
AO	A_o	2.1	D	32.5		INIT1	Principal moments of inertia of telescope outer gimbal (kg-m^2)
ARAY			D			OUTPLF	Final Plot Variable Array
ARAY1			D			OUTPLF	Final X Variable Plot Array
ARAY2			D			OUTPLF	Final Y Variable Plot Array
AT	A_T	2.1	D	2309.		INIT1	Principal moments of inertia of telescope (Kg - m^2)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
ATBAR	$\bar{A}_T$	2.1	D	Computed		INIT2	Moment of inertia of telescope for use in spacecraft dynamics equations ($kg\text{-}m^2$)
ATILT	$\tilde{A}_T$	2.1	D	Computed		INIT2	Moment of inertia of telescope for use in telescope dynamics equations ($kg\text{-}m^2$)
AX			R	0		INIT	Polynomial coefficients for line-of-sight x component.
AY			R	0		INIT	Polynomial coefficients for line-of-sight y component.
AZ			R	0		INIT	Polynomial coefficients for line-of-sight z component.
A0	a_o	2.3	D	Calculated		INIT1	Fine tracking system transfer function coefficient
A1	a_1	2.3	D	Calculated		INIT1	Fine tracking system transfer function coefficient
A2	a_2	2.3	D	Calculated		INIT1	Fine tracking system transfer function coefficient
A3	a_3	2.3	D	Calculated		INIT1	Fine tracking system transfer function coefficient
A1	$(\frac{\ell_2}{\ell_2})$	2.2.2.3	D	28/6		INIT1	Ratio of focal lengths of lenses L_2 to L_1.
A2	$(\frac{\ell_2}{\ell_1}L1)$	2.2.2.3	D	(28/6)(609.601)		INIT1	Ratio of focal lengths of ℓ_2 to ℓ_1 multiplied by f/15 focal plane
A11	A_1'	2.1	D			SCATT	Cross coupling torque acting on spacecraft due to CMG and Experimental Package
A12	A_2'	2.1	D			SCATT	Cross-coupling torque acting on spacecraft due to CMG and Experimental Package

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
A13	A'_3	2.1	D			SCATT	Cross-coupling torque acting on spacecraft due to CMG and Experimental Package
A21		2.1	D			SCATT	Cross-coupling torque acting on spacecraft due to CMG and Experimental Package
A22	A^2_2	2.1	D			SCATT	
A23	A^2_3	2.1	D			SCATT	
A31	A^3_1	2.1	D			SCATT	
A32	A^3_2	2.1	D			SCATT	Cross-coupling torque acting on spacecraft due to CMG and Experimental Package
A33	A^3_3	2.1	D			SCATT	
A41	A^4_1	2.1	D			SCATT	
A42	A^4_2	2.1	D			SCATT	
A43	A^4_3	2.1	D			SCATT	
B	B	2.6	D		Computed	DIRCOS	Transformation matrix from an arbitrary rotating coordinate system to an inertial system (in the LASIM program, B always stands for T2I).
B(I,J)	b_{ij}	2.6	D		Computed	DIRCOS	Elements of B matrix
B(1)	B_1	2.1	D		Computed	SCATT	
B(2)	B_2	2.1	D		Computed	SCATT	Net torque on spacecraft (n-m)
B(3)	B_3	2.1	D		Computed	SCATT	

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
BA	B_a	2.1	D	3.25		INIT1	Principal moment of inertia of CMG outer gimbal (kg-m^2)
BB	B_b	2.1	D	1.68		INIT1	Principal moment of inertia of CMG inner gimbal (kg-m^2)
BETA(1)	β_1	2.1	D	0		SCATT	Inner gimbal angle of CMGs 1, 2, 3 (rads)
BETA(2)	β_2	2.1	D	0		SCATT	
BETA(3)	β_3	2.1	D	0		SCATT	
BG	B_g	2.1	D	1.76		INIT1	Principal moment of inertia of CMG gyro rotor (kg-m^2)
BI	B_I	2.1	D	3.5		INIT1	Principal moment of inertia of CMG gyro rotor (kg-m^2)
BLOOP			I	1		INIT	Start of New Mission Loop
BO	B_o	2.1	D	130.2		INIT1	Principal moment of inertia of telescope outer gimbal (kg-m^2)
BT	B_T	2.1	D	2111.		INIT1	Principal moment of inertia of telescope (kg-m^2)
BTBAR	$\bar{B}_T$	2.1	D	Computed		INIT2	Moment of inertia of telescope for use in spacecraft dynamics equations (kg-m^2)
BTILT	$\tilde{B}_T$	2.1	D	Computed		INIT2	Moment of inertia of telescope for use in telescope dynamics equations (kg-m^2)
BO	b_o	2.3	D	Calculated		INIT1	Fine tracking system transfer function coefficient

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
B1	b_1	2.3	D	Calculated		INIT1	Fine tracking system transfer function coefficient
B2	b_2	2.3	D	Calculated		INIT1	Fine tracking system transfer function coefficient
B2I	[OFF]	2.6	D	Computed		INIT2	Matrix which relates the actual initial telescope frame to the nominal initial telescope frame
B2I	[B2I]		D				Transformation matrix from body to inertial coordinates (note that this matrix is not required in the LASIM program and is not computed).
B3	b_3	2.3	D	Computed		INIT1	Fine tracking system transfer function coefficient
B4	b_4	2.3	D	Computed		INIT1	
B5	b_5	2.3	D	Computed		INIT1	
B6	b_6	2.3	D	Computed		INIT1	
B4	B_4	2.1	D	Computed		SCATT	Cross-coupling torques acting along each CMG outer and inner gimbal axis (n-m).
B5	B_5	2.1	D	Computed		SCATT	
B6	B_6	2.1	D	Computed		SCATT	
B7	B_7	2.1	D	Computed		SCATT	
B8	B_8	2.1	D	Computed		SCATT	
B9	B_9	2.1	D	Computed		SCATT	

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
CA	C_a	2.1	D	13.02		INIT1	Principal moment of inertia of CMG outer gimbal ($kg-m^2$)
CB	G_b	2.1	D	0.35		INIT1	Principal moment of inertia of CMG inner gimbal ($kg-m^2$)
CHAR			1		0	CHKCRD	Test word used for character check
CI	C_I	2.1	D	15.0		INIT1	Principal moment of inertia of telescope inner gimbal ($kg-m^2$)
CMGCLM	CMGCLM	2.1	D	408		INIT1	Vehicle control law control torque magnitude limit (n-m)
CMGPLM	CMGPLM	2.1	D	1500		INIT1	Vehicle control law position error torque magnitude limit (n-m)
CMIN1			D	cos(1')		INIT	Cosine of 1 minute
CMIN30			D	cos(30')		INIT	Cosine of 30 minutes
CNTAB			I		0	CHKCRD	Table containing control words
CO	C_o	2.1	D	119.3		INIT1	Principal moment of inertia of telescope outer gimbal ($kg-m^2$)
COMMA			I	7		CHKCRD	Constant ,
COPY			I	1		INIT	Counter for additional copies of output
CPSI1	$\cos(\psi_1)$		D		0	TELCON	Trigonometric cosine of ψ_1
CPSI2	$\cos(\psi_2)$		D		0	TELCON	Trigonometric cosine of ψ_2

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
CSW				0		INIT1	Control switch (1 if control input)
CT	C_T	2.1	D	954.		INIT1	Principal moment of inertia of telescope (kg-m^2)
C1	C_{1j}	2.6	D		None	DIRCOS	Component of Runge-Kutta parameter, $\underline{k}_{1j}$
C2	C_{2j}	2.6	D		None	DIRCOS	Component of Runge-Kutta parameter, $\underline{k}_{2j}$
C3	C_{3j}	2.6	D		None	DIRCOS	Component of Runge-Kutta parameter, $\underline{k}_{3j}$
CO	C_o	2.3	D	Calcualted		INIT1	Fine tracking system difference equation coefficient
C1	C_1	2.3	D	Calculated		INIT1	
C2	C_2	2.3	D	Calculated		INIT1	
C3	C_3	2.3	D	Calculated		INIT1	
C4	C_4	2.3	D	Calculated		INIT1	
C5	C_5	2.3	D	Calculated		INIT1	
C6	C_6	2.3	D	Calculated		INIT1	
C10D	C_{10d}	2.1	D	Computed		INIT2	Coefficients for vehicle control difference equation representation.
C10DZ	C_{10dz}	2.1	D	Computed		INIT2	

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
C1OP	C_{1op}	2.1	D	Computed		INIT2	Coefficients for vehicle control law difference equation representation.
C1OPZ	C_{1opz}	2.1	D	Computed		INIT2	
C40	C_{40}	2.1	D	Computed		INIT2	Coefficients for CMG velocity loop difference equation representation
C41	C_{41}	2.1	D	Computed		INIT2	
C50	C_{50}	2.1	D	Computed		INIT2	
C51	C_{51}	2.1	D	Computed		INIT2	
DA	A	2.3	D	10^6		INIT1	Fine tracking system, servo amp open loop gain.
DA	DA	2.6	D		None	DIRCOS	Derivative of B matrix computed in DIRCOS (in LASIM program, DA always stands for DT2I).(sec^{-1})
DA(I,J)	DA_{ij}	2.6	D		None	DIRCOS	Element of DA matrix (sec^{-1})
DALPH(1)	$\dot{\alpha}_1$	2.1	D		0	SCATT	Outer gimbal angle rate of CMGs 1,2,3 (rad/sec)/
DALPH(2)	$\dot{\alpha}_2$	2.1	D		0	SCATT	
DALPH(3)	$\dot{\alpha}_3$	2.1	D		0	SCATT	
DALPHX	$\dot{\alpha}_x$	2.4	D		Computed	CONTRL TELCON	Component of inertial angular rate of telescope about $\dot{x}_T$ (rad/sec)
DALPHY	$\dot{\alpha}_y$	2.4	D		Computed	CONTRL TELCON	Component of inertial angular rate of telescope about y_T (rad/sec)

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
DALP1C	$\dot{\alpha}'_{c1}, \dot{\alpha}_{c1}$	2.1	D	Com	Computed	SCATT	
DALP2C	$\dot{\alpha}'_{c2}, \dot{\alpha}_{c2}$	2.1	D		Computed	SCATT	Commanded CMG outer gimbal angle rates (primed-before limiting, unprimed - after limiting) (rad/sec).
DALP3C	$\dot{\alpha}'_{c3}, \dot{\alpha}_{c3}$	2.1	D		Computed	SCATT	
DALP1E	$\dot{\alpha}_{\varepsilon 1}$	2.1	D		Computed	SCATT	
DALP2E	$\dot{\alpha}_{\varepsilon 2}$	2.1	D		Computed	SCATT	CMG outer gimbal angle rate errors (volt)
DALP2E	$\dot{\alpha}_{\varepsilon 3}$	2.1	D		Computed	SCATT	
DA3	A_3	2.3	D	2.67×10^4		INIT1	Fine tracking system, lens drive motor gain
DB	B	2.3	D	141.4		INIT1	Fine tracking system, lens drive motor time constant
DBETA(1)	$\dot{\beta}_1$	2.1	D		0	SCATT	
DBETA(2)	$\dot{\beta}_2$	2.1	D		0	SCATT	Inner gimbal angle rate of CMGS 1,2,3 (rad/sec)
DBETA(3)	$\dot{\beta}_3$	2.1	D		0	SCATT	
DBET1C	$\dot{\beta}'_{c1}$, c1	2.1	D		0	SCATT	
DBET2C	$\dot{\beta}'_{c2}$, c2	2.1	D		0	SCATT	Commanded CMG inner gimbal angle rates (primed-before limiting, unprimed-after limiting)(rad/sec)
DBET3C	$\dot{\beta}'_{c3}, \dot{\beta}_{c3}$	2.1	D		0	SCATT	

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
DBET1E	$\dot{\beta}_{\varepsilon 1}$	2.1	D		Computed	SCATT	CMG inner gimbal angle rate errors (volt)
DBET1E	$\dot{\beta}_{\varepsilon 2}$	2.1	D		Computed	SCATT	
DBET3E	$\dot{\beta}_{\varepsilon 3}$	2.1	D		Computed	SCATT	
DDALPH(1)	$\ddot{\alpha}_1$	2.1	D		0	SCATT	Outer gimbal angle acceleration of CMGs 1,2,3 (rad/sec^2)
DDALPH(2)	$\ddot{\alpha}_2$	2.1	D		0	SCATT	
DDALPH(3)	$\ddot{\alpha}_3$	2.1	D		0	SCATT	
DDBETA(1)	$\ddot{\beta}_1$	2.1	D		0	SCATT	Inner gimbal angle acceleration of CMGs 1,2,3 (rad/sec^2)
DDBETA(2)	$\ddot{\beta}_2$	2.1	D		0	SCATT	
DDBETA(3)	$\ddot{\beta}_3$	2.1	D		0	SCATT	
DDPSI	$\ddot{\psi}_1$	2.1	D		0	CONTL2	Telescope outer gimbal angle acceleration (rad/sec^2)
DDPSI2	$\ddot{\psi}_2$	2.1	D		0	CONTL2	Telescope inner gimbal angle acceleration (rad/sec^2)
DELT(1) D	δ_1	2.1	D		Computed	SCATT	Components of $\underline{\delta}^T$ in the body coordinate system (m)
DELT(2)	δ_2	2.1	D		Computed	SCATT	
DELT(3)	δ_3	2.1	D		Computed	SCATT	

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
DELTAT	T	2.1	D	0.08		INIT1	Distance from center of mass of telescope inner gimbal to center of mass of telescope (m)
DKO	KO	2.3	D	$\frac{1}{240K}$		INIT1	Fine tracking system, voltage divider gain
DK1	K1	2.3	D	.004		INIT1	Fine tracking system, dynamics gain
DK2	K2	2.3	D	141.4		INIT1	Fine tracking system, dynamics time constant
DK3	K3	2.3	D	$\frac{1}{60}$		INIT1	Fine tracking system, velocity sensor gain
DK4	K4	2.3	D	240		INIT1	Fine tracking system, diff. amp break frequency 7 sec.
DLX	$\dot{t}_x$		D		0	FINE	Transfer lens rate - X component (cm/sec)
DLY	$\dot{t}_y$		D		0	FINE	Transfer lens rate - Y component (cm/sec)
DP	P	2.1	D		0	SCATT	X_B component of angular acceleration of main body (rad/sec^2)
DPSI1	$\dot{\psi}_1$	2.1	D		0	CONTL2	Telescope outer gimbal angle rate (rad/sec)
DPSI2	$\dot{\psi}_2$	2.1	D		0	CONTL2	Telescope inner gimbal angle rage (rad/sec)
DQ	$\dot{Q}$	2.1	D		0	SCATT	y_B component of angular acceleration of main body (rad/sec^2)

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
DR	R	2.1	D		0	SCATT	Z_B component of angular acceleration of main body (rad/sec^2)
DRC			1	0		INIT	Flag inhibits control system
DRF			1	0		INIT	Flag inhibits fine tracking system
DRS			1	0		INIT	Flag inhibits spacecraft attitude control system
DRT			1	0		INIT	Flag inhibits telescope control system
DSPACE			1	0		INIT	Flag (1 causes double spacing)
DSW				0		INIT	Data switch (1 if data input)
DT1	T_1	2.3	D	$\frac{1}{2\pi(.466)}$		INIT1	Fine tracking system, lead-lag net time constant
DT2	T_2	2.3	D	$\frac{1}{2\pi(7)}$		INIT1	Fine tracking system, lead-lag net time constant
DT2I	$\frac{d}{dt}[T2I]$	2.6	D		0	DIRCOS	Derivative of [T2I] with respect to time (sec^{-1})
DT4	T_4	2.3	D	$\frac{1}{2\pi(780)}$	0	INIT1	Fine tracking system, servo amp feedback time constant
DUMY						EXEC	Dummy plot array pointer

LASIM Program Dictionary (Continued)

[illegible] NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
D0	d_o	2.3	D	Calculated		INIT1	
D1	d_1	2.3	D	Calculated		INIT1	
D2	d_2	2.3	D	Calculated		INIT1	
D3	d_3	2.3	D	Calculated		INIT1	Fine tracking system difference equation coefficient
D4	d_4	2.3	D	Calculated		INIT1	
D5	d_5	2.3	D	Calculated		INIT1	
D6	d_6	2.3	D	Calculated		INIT1	
D11	d_{11}	2.3	D	Calculated		INIT2	Coefficients for vehicle control law.difference equation representation
D31	d_{31}	2.3	D	Calculated		INIT2	
D32	d_{32}	2.3	D	Calculated		INIT2	
D41	d_{41}	2.1	D	Computed		INIT2	Coefficient for CMG velocity loop difference equation representation
EEX	e_x	2.2.2.5	D		0	CALC2	Transfer lens servo error voltage on X channel (volts)
EEY	e_y	2.2.2.5	D		0	CALC2	Transfer lens servo error voltage
END				0		INIT	Flag signals end of job
EPX	$(70.4)p_x$		D		0	FINE	Total tracking error in X direction (arc seconds)
EPY	$(70.4)p_y$		D		0	FINE	Total tracking error in Y direction (arc seconds)

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
ERV	R_E	2.7	R		.2087E8	ORBGEN	Earth's radius
EX	e_x	2.2.2.3	D		0	FINE	Fine tracking system, X channel error voltage, (volts)
EY	e_y	2.2.2.3	D		0	FINE	Fine tracking system, Y channel error voltage, (volts)
F	K_o	2.2	D	7.92×10^{-10}		INIT1	Optical system gain
FLAG			1		0	INIT	Fine control system processing switch
FOUNDA				0		INIT	Flag (1 of data input)
FRAC	μ	2.2.2.3	D	0	0	XYCURV	Energy fraction, $\frac{E_p}{E_n}$ (dimensionless)
GCC	G_{cc}	2.1	D	2.24		INIT1	Cross-compensation gain in CMG velocity loop (volt/volt)
GCMG1	G_1	2.1	D	1.3436×10^7		INIT1	Gain of vehicle pitch and yaw control laws (n-m/rad)
GCMG1Z	G_{1z}	2.1	D	1.68×10^6		INIT1	Gain of vehicle law (n-m/rad)
GCMG2	G_{xmg2}	2.1	D	56		INIT1	Gain on actual CMG angular rate for use in computing rate errors (v/rad/sec)
GCMG3	G_{cmg3}	2.1	D	859.95		INIT1	Gain of CMG summing amp, motor, and gear (n-m/v)
H			D	.01	0	INIT	Basic time step in milliseconds as used in program

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
H	T_S	2.1	D	0.01		INIT1	Time between evaluations of spacecraft and CMG dynamics and control equations(sec)
HC		2.1	D	.01	0	INIT	Basic time step in milliseconds as used in program
HC	T_T	2.1	D	0.01		INIT1	Time between evaluations of telescope control and dynamics equations (sec)
HC	H_c	2.4	D	0.01			Time interval between evaluations of telescope control and dynamics equations and computation of [T2I] (sec).
HEXB	H_{EXB}	2.1	D		Computed	SCATT	Difference between commanded and actual CMG momentum (n-m-sec)
HEYB	H_{EyB}	2.1	D		Computed	SCATT	
HEZB	H_{EzB}	2.1	D		Computed	SCATT	
HF			D	.002	0	INIT	Fine time step in milliseconds as used in program
HF	T	2.1	D	0.002		INIT1	Time between evaluations of fine system control equations (sec)
HXBA	H_{XBA}	2.1	D		Computed	SCATT	Component of actual CMG momentum in body coordinates(n-m-sec)
HXBC	H_{XBC}	2.1	D		Computed	SCATT	Component of total CMG momentum command in body system (n-m-sec)
HYBA	H_{YBA}	2.1	D		Computed	SCATT	Component of actual CMG momentum in body coordinates (n-m-sec)

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
HYBC	H_{YBC}	2.1	D		Computed	SCATT	Component of total CMG momentum command in body system (n-m-sec)
HZBA	H_{ZBA}	2.1	D		Computed	SCATT	Component of actual CMG momentum in body coordinates (n-m-sec)
HZBC	H_{ZBC}	2.1	D		Computed	SCATT	Component of total CMG momentum command in body system (n-m-sec)
H1	H_1	2.1	D	2720		INIT1	Angular momentum of CMG1 rotor $(=\Omega_1 A_g)$(n-m-sec)
H2	H_2	2.1	D	2720		INIT1	Angular momentum of CMG2 rotor $(=\Omega_2 A_g)$(n-m-sec)
H3	H_3	2.1	D	2720		INIT1	Angular momentum of CMG3 rotor $(=\Omega_3 A_g)$(n-m-sec)
I	E_T	2.2.2.3	D	10"		TELCON	Total light power incident upon f/70 focal plane (photons/sec)
IER						INIT	Error message pointer
INPUT				5		INIT	Input Tape
INT	E_T	2.2	D		10"	FINE	Total light power (photons/sec)
IOP			I		0	PROCON	Integer operand
IPLEX			I	-1		INIT	Plot variable counter
IPOINT						CHKCRD	Processing pointer used in PROCON
ISYM			I			OUTPLF	Plotting symbol
IXX	I_{xx}	2.1	D	Computed		INIT2	Component of inertia dyadic $\square_v$ in body coordinate system (kg-m^2)

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
IXXP	I_{xx}'	2.1	D	371473		INIT1	Moment and product of inertia of spacecraft (main body) in body coordinates about mass centroid of main body (kg-m^2)
IXY	I_{XY}	2.1	D	Computed		INIT2	Component of inertia dyadic $\square_v$ in body coordinate system (kg-m^2)
IXYP	I_{XY}'	2.1	D	0		INIT1	Moment and product of inertia of spacecraft (main body) in body coordinates about mass centroid of main body (kg-m^2)
IXZ	I_{XZ}	2.1	D	Computed		INIT2	Component of inertia dyadic $\square_v$ in body coordinate system (kg-m^2)
IXZP	I_{XZ}'	2.1	D	0		INIT1	Moment and product of inertia of spacecraft (main body) in body coordinates about mass centroid of main body (kg-m^2)
IYY	I_{YY}	2.1	D	Computed		INIT2	Component of inertia dyadic $\square_v$ in body coordinate system (kg-m^2)
IYYP	I_{YY}'	2.1	D	341473		INIT1	Moment and product of inertia of spacecraft (main body in body coordinates about mass centroid of main body (kg-m^2)
IYZ	I_{YZ}	2.1	D	Computed		INIT2	Component of inertia dyadic $\square_v$ in body coordinate system (kg-m^2)
IYZP	I_{YZ}'	2.1	D	0		INIT1	Moment and product of inertia of spacecraft (main body) in body coordinates about mass centroid of mass body (kg-m^2)

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
IZZ	I_{ZZ}	2.1	D	Computed		INIT2	Component of inertia dyadic $\|\|v_2$ in body coordinate system (kg-m^2)
IZZP	I'_{ZZ}	2.1	D	42730		INIT1	Moment and product of inertia of spacecraft (main body) in body coordinates about mass centroid of main body (kg-m^2)
T2I	[T2IN]	2.6	D	Computed		INIT2	[T2I] matrix for nominal initial telescope attitude ($\alpha_x = \quad_y = 0$)
JBCARD			I			INIT	Buffer for page heading
JMR	J_{mr}	2.1	D	0.006888		INIT1	Moment of inertia of CMG torque motor rotor about spin axis (kg-m^2)
JPRCNT			I	0		INIT	Counter - Counts no. of words requested for printing
JSW			I	0		INIT	Switch (1 indicator of card already received)
KF	K_f	2.1	D	17.0		INIT1	Flexure spring rate of telescope flex-pivot gimbals (n-m/rad)
K1K	K_f	2.1	I	50		INIT	Print frequency parameter
K1KP			I	10		INIT	Plot frequency parameter
KSL	K_{SL}	2.1	D	0.00257		INIT1	H vector control law gain (rad/sec/n-m)
KWTA	K_{WTA}	2.1	D	3.0		INIT1	Gain in CMG outer gimbal velocity loop (volt/volt)
KWTB	K_{WTB}	2.1	D	2.44		INIT1	Gain in CMG inner gimbal velocity loop (volt/volt)

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
K3	K_3	2.4	D	2740		INIT1	Gain of telescope control rate loop (volt/rad/sec)
LAD	λ_g	2.7	I	32		INIT	Latitude of the groudn station (degrees)
LCNT			I	0		INIT	Counter - Counts print lines per page
LINCNT			I	60		INIT	Specifies maximum no. of print lines per page
LIST				0		INIT	Pre-list flag (1 causes pre-list)
LL			I	-1		OUTPLF	Plot flag for point connection
LMAG	$\|L_S\|$	2.2	D			EXEC	Magnitude of line-of-sight vector (meters)
LMX	L_{mx}	2.6	D		Computed	EXEC	x component in inertial system of $\underline{L}_M$
LMY	L_{my}	2.6	D			EXEC	y component in inertial system of $\underline{L}_M$
LMZ	L_{mz}	2.6	D		Computed	EXEC	z component in inertial system of $\underline{L}_M$
LOD	O_g	2.7	I		-104	INT	Longitude of the ground station degrees
LPLS			I	0		INIT	Counter - Counts no. of words requested for plotting
LPRS			I		24	OUTPRT	Counter - Counts number of variables requested for printing

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
LX	L_x	2.2.2.3	D			FINE	X component of line-of-sight vector in [T] frame
LY	L_y	2.2.2.3	D			FINE	Y component of line-of-sight vector in [T] frame
LZ	L_z	2.2.2.3	D			FINE	Z component of line-of-sight vector in [T] frame
M	M^o	2.1	D	24494.		INIT1	Mass of main body (kg)
MA1	$M\alpha_1$	2.1	D		Computed	SCATT	Torque produced by CMG1, 2, 3 outer gimbal torques after reflection through gearing (n-m)
MA2	$M\alpha_2$	2.1	D		Computed	SCATT	
MA3	$M\alpha_3$	2.1	D		Computed	SCATT	
MB1	$M\beta_1$	2.1	D		Computed	SCATT	Torque produced by CMG1, 2, 3 inner gimbal-torquer after reflection through gearing (n-m)
MB2	$M_{\beta 2}$	2.1	D		Computed	SCATT	
MB3	$M_{\beta 3}$	2.1	D		Computed	SCATT	
MEX	M_{EX}	2.1	D		Computed	SCATT	X_B component of $\underline{M}^E$ (newtons)
MEY	M_{EY}	2.1	D		Computed	SCATT	Y_B component of $\underline{M}^E$ (newtons)
MEZ	M_{EZ}	2.1	D		Computed	SCATT	Z_B component of $\underline{M}^E$ (newtons)
MT1	M_{T1}	2.1	D		0	TELCON	Output of telescope outer gimbal torque motor after limiting (n-m)
MT2	M_{T2}	2.1	D		0	TELCON	Output of telescope inner gimbal torque motor after limiting (n-m)

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
M1	M^1	2.1	D	0		INIT1	Mass of CMG1 outer gimbal (kg)
M2	M^2	2.1	D	0		INIT1	Mass of CMG1 inner gimbal (kg)
M3	M^3	2.1	D	0		INIT1	Mass of CMG1 gyro rotor (kg)
M4	M^4	2.1	D	0		INIT1	Mass of CMG2 outer gimbal (kg)
M4	M	2.1	D	Computed		INIT2	Mass of entire LASIM vehicle, including all parts (kg)
M5	M^5	2.1	D	0		INIT1	Mass of entire LASIM vehicle, including 2 inner parts (kg)
M6	M^6	2.1	D	0		INIT1	Mass of entire LASIM vehicle, including 2 gyro rotor (kg)
M7	M^7	2.1	D	0		INIT1	Mass of entire LASIM vehicle, including 3 outer gimbal (kg)
M8	M^8	2.1	D	0		INIT1	Mass of entire LASIM vehicle, including 3 inner gimbal (kg)
M9	M^9	2.1	D	0		INIT1	Mass of entire LASIM vehicle, including 3 gyro rotor (kg)
M10	M^{10}	2.1	D	0		INIT1	Mass of telescope outer gimbal (kg)
M11	M^{11}	2.1	D	0		INIT1	Mass of telescope inner gimbal (kg)
M12	M^{12}	2.1	D	2000		INIT1	Mass of telescope (kg)

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
M12T	$\tilde{M}^{12}$	2.1	D	Computed		INIT2	Reduced mass of telescope
NAME			I			Block Data	Table contains output names
NBE			I			PVINO	Eccentric anomaly estimate
NFINE			I		0	INIT	Fine loop counter
NG	N_G	2.1	D	56		INIT1	
NLOOP			I	1000		INIT	End of new mission loop
NLOOPF			I	5		INIT	End of fine loop
NOF				0		BLOCK DATA	No. of jobs with plot output
NONPRO				0		INIT	Non-processing switch (1 inhibits printing)
NPTS				0		BLOCK DATA	No. of points to be plotted
NV			I		0	OUTPLT	Number of plot variables for a given job
NVAL			I	123		INIT	Specifies no. of output variables
NVAR			I	0		BLOCK DATA	No. of variables to be plotted
O	W_o	2.4	D	60π		INIT1	Telescope rate gyro undamped natural frequency (rad/sec)
OBE	E	2.7	R		0	PVINO	Eccentric anomaly
OBO	Ω	2.7	R	0		INIT	Longitude of the ascending node

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
OBT	O_s	2.7	R		-.104E3	INIT	Initial longitude of the space vehicle (degrees)
OPER	τ	2.7	R	86000		INIT	Period of the orbit
ORA	R_a	2.7	R			INIT	Orbit's apogee altitude
ORP	R_p	2.7	R			INIT	Orbit's perigee altitude
OSA	a	2.7	R			INIT	Semi-major axis of the orbit
OSE	e(e)	2.7	R	.05		INIT	Eccentricity of the orbit
OSI	i	2.7	R	.283E2		INIT	Inclination of orbit plane with respect to earth's equator plane
OUTPUT			I	6		INIT	Output tape
P	P	2.1	D		0	SCATT	X_B component of angular velocity $\underline{\omega}^B$ (rad/sec)
PAGENO			I	0		INIT	Counter - specifies current output page no.
PLBUF			I			OUTPLT	Plot variable buffer
PLBUFS			I			INIT	Specifies size of plot buffer
PLBUFS			I	100		OUTPLT	Plot buffer size
PLOT			I	0		INIT	Flag signals user request for plots
PLOINT			I			PROCON	Plot variable pointer

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
POSEX	POSEX	2.4	D		Computed	TELCON	Position error commands from an ideal coarse or fine optical sensor (volt)
POSEY	POSEY	2.4	D		Computed	TELCON	
POSXLI	POSXL	2.4	D		0	TELCON	Position error commands from actual coarse or fine optical sensor (volt)
POSYLI	POSYL	2.4	D		0	TELCON	
PRBUF1			I			INIT	Buffer contains names of variables to be printed and plotted
PRBUFR			I			OUTPRT	Print variable buffer
PRBUFS			I	23		INIT	Counter - specifies no. of words to be printed
PRINT			I	0		INIT	Flag signals user request for print
PROINT			I			PROCON	Print variable pointer
PSI1	ψ_1	2.1	D		0	CONTL2	Telescope outer gimbal angle (rad)
PSI2	ψ_2	2.1	D		0	CONTL2	Telescope inner gimbal angle (rad).
PX	P_x	2.2.2.3	D		0	CALC2	Image center X coordinate in f/70 plane (cm)
PY	P_y	2.2.2.3	D		0	CALC2	Image center Y coordinate in f/70 plane (cm)
Q	Q	2.1	D		0	SCATT	y_B component of angular velocity $\underline{\omega}^B$ (rad/sec)
QQ			D		0	INIT	Runge-Kutta Gill integration variable

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
QQQ			D			INIT	Runge-Kutta Gill integration variable
QUIT			D	10	0	INIT	Mission time in seconds as used in program
R	R	2.1	D		0	SCATT	Z_B component of angular velocity $\underline{\omega}$ (rad/sec)
RATEXI	RATEXI	2.4	D		Computed	TELCON	Component of measured inertial angular rate of telescope about x_T (rad/sec)
RATEYI	RATEYI	2.4	D		Computed	TELCON	Component of measured inertial angular rate of telescope about y_T (rad/sec)
RATIOX	μ	2.2	D			XYCURV	Energy fraction
RATIOY	μ	2.2	D			XYCURV	Energy fraction
ROAREA						INIT	Read input area
RESTAR				0		INIT	Flag (1 causes restart tape to be read)
RO(1)	ρ_1^0	2.1	D	.0381		INIT1	Components in body system of $\underline{\rho}^0$, distance from 0 to center of mass of body (m)
RO(2)	ρ_2^0	2.1	D	-.0890		INIT1	
RO(3)	ρ_3^0	2.1	D	0		INIT1	
RO10(1)	ρ_1^{10}	2.1	D	.0381		INIT1	Components in body system of $\underline{\rho}^{10}$, distance from 0 to center of mass of telescope gimbal (m)
RO10(2)	ρ_2^{10}	2.1	D	-.089		INIT1	
RO10(3)	ρ_3^{10}	2.1	D	9.9		INIT1	

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
RREST			I	0		INIT	Flag (1 causes restart tape to be created)
RZM	R_s	2.7	R		0	ORBGEN	Initial space vehicle modulus in inertial frame
RZX	X_s	2.7	R		0	ORBGEN	Initial space vehicle x-coordinate in inertial frame
RZY	Y_s	2.7	R		0	ORBGEN	Y-coordinate in inertial frame
RZZ	Z_s	2.7	R		0	ORBGEN	Z-coordinate in inertial frame
SA(I,J)	a_{ij}	2.1	D		Computed	SCATT CONTRL	Matrix of inertias used for CMG, telescope, and spacecraft dynamics equations (I=1,...,11; j=1,...,11)($kg\text{-}m^2$)

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
S_{B1}	b_1	2.1	D		Computed	SCATT	Cross-coupling torques acting on spacecraft due to spacecraft, CMG's, telescope, and CMG torque motor rotors (n-m).
S_{B2}	b_2	2.1	D		Computed	SCATT	
S_{B3}	b_3	2.1	D		Computed	SCATT	
SB10	b_{10}	2.1	D		Computed	CONTRL	Cross-coupling torques acting along telescope outer and inner gimbal axes (n-m).
SB11	b_{11}	2.1	D		Computed	CONTRL	
SD1	L_x	2.2,2.3	D			FINE	X component of line-of-sight vector in [T] frame.
SD2	L_y	2.2,2.3	D			FINE	Y component of line-of-sight vector in [T] frame.
SD3	L_z	2.2,2.3	D			FINE	Z component of line-of-sight vector in [T] frame.
SEQ					0	INIT	Card sequence counter.
SHIFT			I		0	CHKCRD	Test word used for comma check.
SIG	ζ	2.4	D	0.7		INIT1	Telescope rate gyro damping ratio.
SPSI1	$\sin(\psi_1)$		D		0	TELCON CONTL2	Trignometric sine of ψ_1.
			D		0		
SPSI2	$\sin(\psi_2)$		D		0	TELCON CONTL2	Trignometric sine of ψ_2.
SLASHA				/*		BLOCK DATA	Flag Signals End of Batch.
SRXMU	μ	2.7	R	.118644E9		ORBGEN	Gravitational constant.
SSWP	ω'	2.7	R		0	ORBGEN	Argument of perifocus.

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
SUBS			I			OUTPRT	Subscripts for printing.
TAO	T_{ao}	2.4	D	Computed		INIT2	Coefficients of difference equation representation of telescope position sensor lag.
'TA2	T_{a1}	2.4	D	Computed		INIT2	
TA2	T_{a2}	2.4	D	Computed		INIT2	
TCMG1	τ_1	2.1	D	0.225		INIT1	Time constant for differentiator in vehicle control law (sec.)
TCMG2	τ_2	2.1	D	0.0053		INIT1	Time constant for first order lag in vehicle control law (sec.)
TCMG3	τ_3	2.1	D	0.05		INIT1	Time constants in CMG velocity loop (sec.).
TCMG4	τ_4	2.1	D	5.0		INIT1	
TCO	T_{co}	2.4	D	Computed		INIT2	Coefficients of difference equations representation of telescope rate gyros.
TC1	T_{c1}	2.4	D	Computed		INIT2	
TC2	T_{c2}	2.4	D	Computed		INIT2	
TDO	T_{do}	2.4	D	Computed		INIT2	Coefficients of difference equation representation of telescope rate gyros.
TD1	T_{d1}	2.4	D	Computed		INIT2	
TD2	T_{d2}	2.4	D	Computed		INIT2	
TESTWD	-	-	I	-	0	CHKCRD	Temporary storage word.
TESTWD	-	-	I	-	0	PROCON	Temporary storage.
TESTWO1	-	-	I	-	0	PROCON	Temporary storage.
TESTWO2	-	-	I	-	0	PROCON	Temporary storage.

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
THETAZ	θ_z	2.2	D	-	0	FINE	Angle between telescope longitudinal axis and line-of-sight.
THS	θ_s	2.7	R	-	0	ORBGEN	True anomaly.
TH1	T_{H1}	2.6	D	-	NONE	DIRCOS	Intermedate parameter in [T2I] integration (rad.).
TH2	T_{H2}	2.6	D	-	NONE	DIRCOS	Intermediate parameter in [T2I] integration (rad.).
TH3	T_{H3}	2.6	D	-	NONE	DIRCOS	Intermediate parameter in [T2I] integration (rad.).
TITLE	-	-	I	-	-	INIT	Buffer for output title.
TLX	t_x	2.2,2.3	D	-	0	CALC2	Transfer lens, X position coordinate (cm).
TLY	t_y	2.2,2.3	D	-	0	CALC2	Transfer lens, Y position coordinate (cm).
TM	M	2.7	R	-	0	PVINO	Mean anomaly.
TMA1	$T_{M_{\alpha_1}}$	2.1	D	-	0	SCATT	Intermediate parameters in CMG velocity loop (Volt).
TMA2	$T_{M_{\alpha_2}}$	2.1	D	-	0	SCATT	
TMA3	$T_{M_{\alpha_3}}$	2.1	D	-	0	SCATT	
TMB1	$T_{M_{\beta_1}}$	2.1	D	-	0	SCATT	Intermediate parameters in CMG velocity loop (volt).
TMB2	$T_{M_{\beta_2}}$	2.1	D	-	0	SCATT	

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
TMB3	$T_{M_{\beta_3}}$	2.1	D	-	0	SCATT	Intermediate parameters in CMG velocity loop (volt).
TORQX	τ'_x, τ_x	2.1	D	-	Computed	SCATT	Vehicle control law control torque commands.
TORQXD	τ_{xd}	2.1	D	-	0	SCATT	Vehicle control law rate damping torque commands (n-m).
TORQXP	τ'_{xp}, τ_{xp}	2.1	D	-	Computed	SCATT	Vehicle control law position error torque command.
TORQY	τ'_y, τ_y	2.1	D	-	Computed	SCATT	Vehilce control law control torque command.
TORQYD	τ_{yd}	2.1	D	-	0	SCATT	Vehicle control law rate damping torque command (n-m).
TORQYP	τ'_{yp}, τ_{yp}	2.1	D	-	Computed	SCATT	Vehilce control law position error torque command.
TORQXD	τ_{zd}	2.1	D	-	0	SCATT	Vehicle control law rate damping torque commands (n-m).
TORQZP	τ'_{zp}, τ_{zp}	2.1	D	-	Computed	SCATT	Vehicle control law position error torque command.
TP2CNT	-	-	I	0	-	INIT	Counter - Counts print records on WATP2.
TTX	t_x	2.2,2.5	D	0	0	FINE	Transfer lens X-position coordinate (cm).
TTY	t_y	2.2,2.5	D	0	0	FINE	Transfer lens Y-position coordinate (cm).

. LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
TVM	-	-	I	0	-	INIT	Flag causes orbit generating subprogram to be run.
TWEN	-	-	I	20	0	PROCON	Constant 20, sets maximum number of printout copies.
T1	T_1	2.6	D	0	NONE	DIRCOS	Intermediate parameter in [T2I] integration (rad.).
T2	T_2	2.6	D	0	NONE	DIRCOS	Intermediate parameter in [T2I] integration (rad.).
T2B	[T2B]	2.6	D	-	Computed	CONTL2	Transformation matrix from telescope to body coordinates.
T2I	[T2I]	2.1	D	-	Computed	DIRCOS	Transformation matrix from telescope coordinates to inertial coordinates.
T3	T_3	2.6	D	-	NONE	DIRCOS	Intermediate parameter in [T2I] integration (rad.).
UTVM	-	-	I	0	-	INIT	Flag causes Orbit Generating subprogram data to be used.
VALTP	-	-	I	12,13,14,15	-	PROCON	Table contains alternate tape units.
VLOSX	$\dot{L}_x$	2.7	R	-	0	ORBGEN	Line-of-sight velocity, x-coordinate in inertial frame.
VLOSY	$\dot{L}_y$	2.7	R	-	0	ORBGEN	Line-of-sight velocity, y-coordinate in inertial frame.
VLOSZ	$\dot{L}_z$	2.7	R	-	0	ORBGEN	Line-of-sight velocity, z-coordinate in inertial frame.

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
VX	V_x	2.6	D	-	Computed	INIT2	X component in inertial system of $\underline{V}$ (m/sec).
VY	V_y	2.6	D	-	Computed	INIT2	Y component in inertial system of $\underline{V}$ (m/sec).
VZ	V_z	2.6	D	-	Computed	INIT2	Z component in inertial system of $\underline{V}$ (m/sec).
WKTP1	-	-	-	1	-	INIT	Work Tape 1.
WKTP2	-	-	-	2	-	INIT	Work Tape 2.
WKTP3	-	-	-	3	-	INIT	Work Tape 3.
WKTP4	-	-	-	4	-	INIT	Work Tape 4.
WKTP9	-	-	-	9	-	INIT	Work Tape 9.
WKTP10	-	-	-	10	-	INIT	Work Tape 10.
WKTP11	-	-	-	11	-	INIT	Work Tape 11.
WX	ω_x	2.6	D	-	NONE	DIRCOS	X component of angular velocity of an arbitrary rotating coordinate frame (in the LASIM program, WX always stands for (WX4 + WX4OLD)/2) (rad/sec).
WX1	ω_{1x}	2.1	D	-	0	SCATT	Component of $\underline{\omega}_1$ relative to CMG1 inner gimbal coordinate system (rad/sec).
WX1P	ω'_{1x}	2.1	D	-	0	SCATT	Component of $\underline{\omega}'_1$ relative to CMG1 outer gimbal coordinate system (rad/sec).

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
WX2	ω_{2x}	2.1	D	-	0	SCATT	Component of $\underline{\omega}_2$ relative to CMG2 inner gimbal coordinate system (rad/sec).
WX2P	ω'_{2x}	2.1	D	-	0	SCATT	Component of $\underline{\omega}_2$ relative to CMG2 outer gimbal coordinate system (rad/sec).
WX3	ω_{3x}	2.1	D	-	0	SCATT	Component of $\underline{\omega}_3$ relative to CMG3 inner gimbal coordinate system (rad/sec).
WX3P	ω'_{3x}	2.1	D	-	0	SCATT	Component of $\underline{\omega}'_3$ relative to CMG3 outer gimbal coordinate system (rad/sec).
WX4	ω_{4x}	2.1	D	-	0	CONTL2 CONTRL TELCON	X componet of telescope angular velocity in telescope frame (rad/sec).
WX4P	ω'_{4x}	2.1	D	-	0	CONTRL	Component of $\underline{\omega}'_4$ relative to telescope outer gimbal coordinate (rad/sec).
WX4P	ω'_{4x}	2.1	D	-	0	CONTRL	Component of $\underline{\omega}'_4$ relative to telescope outer gimbal coordinate (rad/sec).
WY	ω_y	2.6	D	-	None	DIRCOS	Y component corresponding to WX (rad/sec).
WY1	ω_{1y}	2.1	D	-	0	SCATT	Component of $\underline{\omega}_1$ relative to CMG1 inner gimbal coordinate system (rad/sec).

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
WY1P	ω'_{1y}	2.1	D	-	0	SCATT	Component of $\underline{\omega}'_1$ relative to CMG1 outer gimbal coordinate system (rad/sec).
WY2	ω_{2y}	2.1	D	-	0	SCATT	Component of $\underline{\omega}_2$ relative to CMG2 inner gimbal coordinate system (rad/sec).
WY2P	ω'_{2y}	2.1	D	-	0	SCATT	Component of $\underline{\omega}'_2$ relative to CMG2 outer gimbal coordinate system (rad/sec).
WY3	ω_{3y}	2.1	D	-	0	SCATT	Component of $\underline{\omega}_3$ relative to CMG3 inner gimbal coordinate system (rad/sec).
WY3P	ω'_{3y}	2.1	D	-	0	SCATT	Component of $\underline{\omega}'_3$ relative to CMG3 outer gimbal coordinate system (rad/sec).
WY4	ω_{4y}	2.1	D	-	0	CONTL2 CONTRL TELCON	Y component of telescope angular velocity in telescope frame (rad/sec).
WY4P	ω'_{4y}	2.1	D	-	0	SCATT	Component of $\underline{\omega}'_4$ relative to telescope outer gimbal coordinates (rad/sec).
WZ	ω_z	2.6	D	-	None	DIRCOS	Z component corresponding to WX (rad/sec).
WZ1	ω_{1z}	2.1	D	-	0	SCATT	Component of $\underline{\omega}_1$ relative to CMG1 inner gimbal coordinate system (rad/sec).
WZ1P	ω'_{1z}	2.1	D	-	0	SCATT	Component of $\underline{\omega}'_1$ relative to CMG1 outer gimbal coordinate system (rad/sec).

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
WZ2	ω_{2z}	2.1	D	-	0	SCATT	Component of $\underline{\omega}_2$ relative to CMG2 inner gimbal coordinate system (rad/sec).
WZ2P	ω'_{2z}	2.1	D	-	0	SCATT	Component of $\underline{\omega}'_2$ relative to CMG2 outer gimbal coordinate system (rad/sec).
WZ3	ω_{3z}	2.1	D	-	0	SCATT	Component of ω_z relative to CMG3 inner gimbal coordinate system (rad/sec)
WZ3P	ω'_{3z}	2.1	D	-	0	SCATT	Component of $\underline{\omega}'_3$ relative to CMG3 outer gimbal coordinate system (rad/sec).
WZ4	ω_{4z}	2.1	D	-	0	CONTL2 CONTRL TELCON	Z Component of telescope angular velocity in telescope frame (rad/sec).
WZ4P	ω'_{4z}	2.1	D	-	0	CONTRL	Component of $\underline{\omega}'_4$ relative to telescope outer gimbal coordinates (rad/sec).
X	-	-	D	0	-	INIT	Accumulated simulation time (sec.).
XARAYX	-	-	R	-	-	EXEC	Line-of-sight X component.
XARAYY	-	-	R	-	-	EXEC	Line-of-sight Y component.
XARAYZ	-	-	R	-	-	EXEC	Line-of-sight Z component.
XCOORD	-	-	I	-	-	OUTPLT	X plot coordinate label.
YARRAYX	-	-	R	-	-	EXEC	Time array.

LASIM Program Dictionary (Continued)

PROGRAM NAME	MATH SYMBOL	VOL. I PARA. WHERE DEFINED	PRECISION	IF CONSTANT, NOMINAL VALUE	IF VARIABLE NOMINAL INITIAL VALUE	SUBROUTINE WHERE ORIGINATED	DESCRIPTION AND UNITS
YCOORD	-	-	I	-	-	OUTPLF	Y plot coordinate label.
ZERO	-	-	D	0	0	INIT	Constant zero.

LASIM Program Dictionary (Continued)

SECTION 6

POINTING CONTROL PROGRAM

The Pointing Control program consists of a separate program deck which is run independently from the "main" or LASIM program. As explained in Section 1, the tracking functions of the LCSE are simulated in the LASIM program described in Paragraphs 1 through 5 of this report. The operations of the LCSE hardware associated with directing the downlink beam, or "pointing" functions are simulated in the Pointing Control program. Definition of the mathematical formulations for the Pointing Control program are contained in Paragraph 2.5 of the Laser Aiming Simulation (LASIM) Final Report, Volume I - Mathematical Formulation, IBM Report No. 68-K10-0006.

The Pointing Control program is written in FORTRAN IV, Version 13 and runs on the IBM 7094 computer described in Section 4. Like the LASIM program, the Pointing Control program is written in double precision. One input tape must be provided for the Pointing Control program. This tape is generated by the LASIM program as described in Paragraphs 6.3.1 and 6.3.2.

The following paragraphs describe the operation and usage of the Pointing Control program.

6.1 PROGRAM FUNCTION AND DESCRIPTION

The Pointing Control Program performs the following functions:

- o Simulate Point-Ahead Ground Computations
- o Simulate Spaceborne Sun Sensor Operation
- o Simulate Risely Prism Servo Operation
- o Compute Total Pointing Errors Including Tracking System Contribution

Contained within the program are the equations which must be solved to represent the ground computations, the sun sensor angles, the Risely prism servoes, and the error computations. The inputs to these equations, however, must be formed from data furnished by the LASIM, or tracking simulation program. The following correlates the required input quantities with the corresponding use to which they are put.

Program Input	Use
o Spacecraft line-of-sight vector	Compute commanded point-ahead angle
o Spacecraft line-of-sight rate vector	Formulate Risely prism servo commands
o T2I transformation matrix	Computed desired pointing vector

Program Input	Use
o Transfer lens position	Compute actual pointing vector from which error computations are made.

The Pointing Control program is a small program by comparison to LASIM. The automatic features included in the LASIM program such as print, plot, input and output options are not included in the Pointing Control program. This program is written in a straightforward manner and because of the small number of hardware functions simulated, hardware parameter values for the Risely prism servoes and the other program variables are set through standard data cards as discussed in Paragraphs 6.3.2 and 6.3.4. No restart capability is provided in the program. The following paragraph describes the operation of the program to accomplish the functions enumerated herein.

6.2 PROGRAM ORGANIZATION

Figure 6-1 illustrates the functional flow through the Pointing Control program. This program has been written in a sequential manner without modularization and allocation of distinct functions to separate subroutines. The groups of calculations related to a particular function are clearly indicated in the listing by comment cards; and the listing closely parallels the flow chart of Figure 6-1.

Execution of the program begins with initialization of program constants and hardware parameter values. Initialization is performed by reading data cards on which the values of program variables likely to change from one run to another are placed. The Risely prism difference equation coefficients are evaluated in this initialization.

After initialization, up to 500 records are read from the input tape into core. The number of records read in, is determined by the total mission time for which simulation is desired. The program variable QUIT is set equal to the mission time; and <u>should be exactly equal</u> to the value used in the LASIM program run from which the input tape being used was generated. Each record contains the data words shown in Table 6-1, in the indicated order. As discussed previously, a record containing the data words shown in Table 6-1 is written each pass through the basic time step loop in the LASIM program.

The Pointing Control program processes the data in each record in a similar basic time step loop. The program variable T is used as the basic time step variable and must be <u>set equal</u> to the basic time step used in the LASIM program run from which the input tape being used was generated.

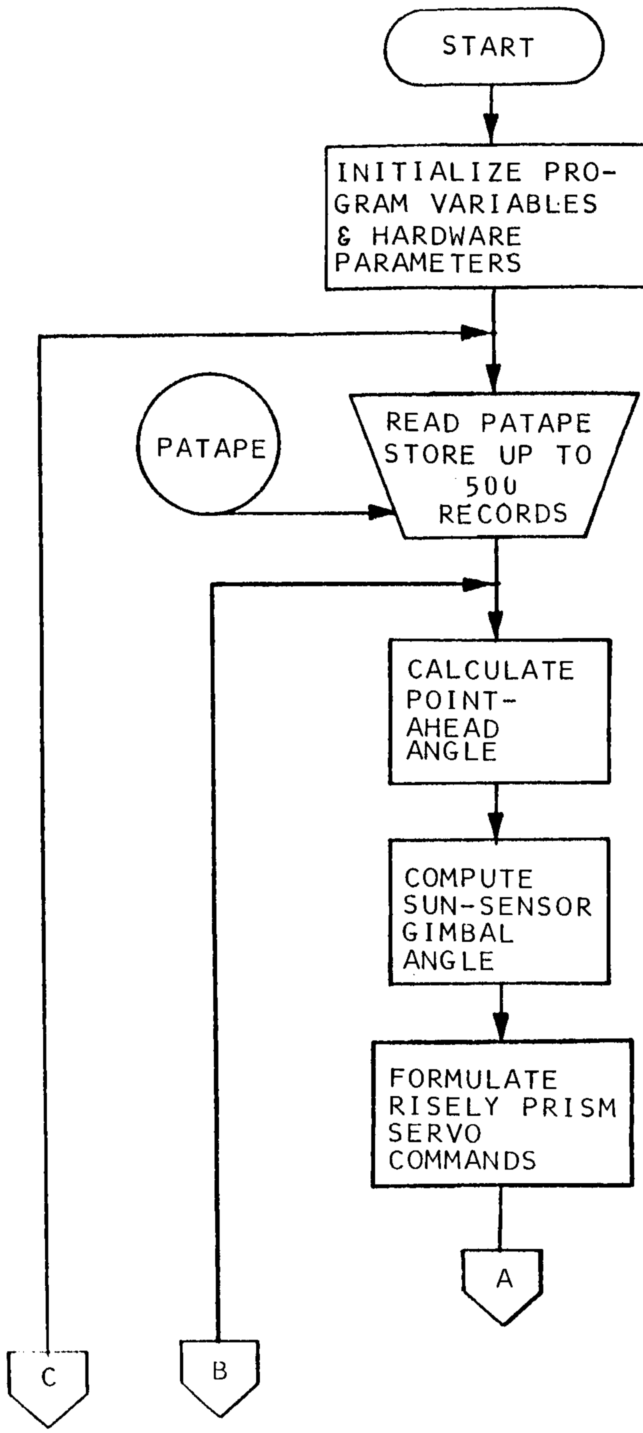


FIGURE 6-1. POINTING CONTROL PROGRAM FUNCTIONAL FLOW

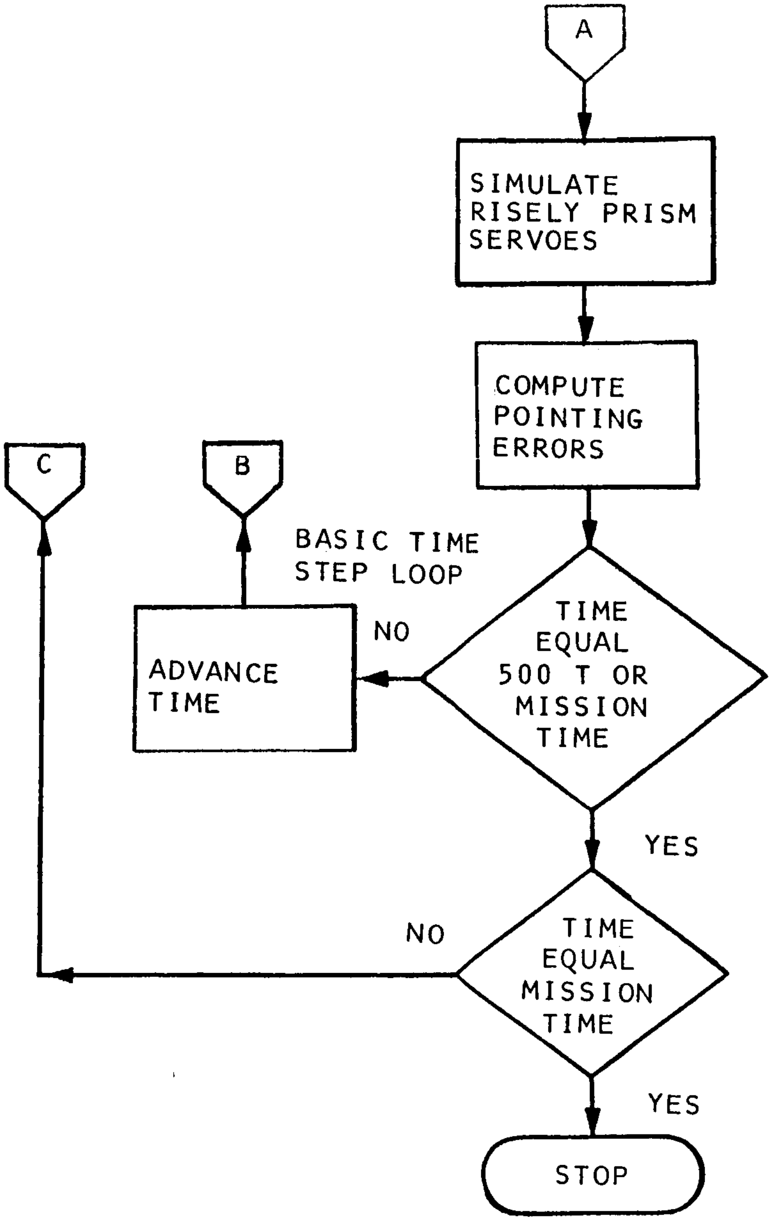


FIGURE 6-1. POINTING CONTROL PROGRAM FUNCTIONAL FLOW (CONTINUED)

TABLE 6-1. POINTING CONTROL TAPE RECORD WORD LIST

Word Number	Program Name	Description
1	LMX	Line-of-sight vector components in inertial coordinates
2	LMY	
3	LMZ	
4	VLMX	Line-of-sight rate vector components in inertial coordinates
5	VLMY	
6	VLMZ	
7	T2I(1,1)	Telescope-to-inertial transformation matrix elements
8	T2I(2,1)	
9	T2I(3,1)	
10	T2I(1,2)	
11	T2I(2,2)	
12	T2I(3,2)	
13	T2I(1,3)	
14	T2I(2,3)	
15	T2I(3,3)	
16	TLX	Transfer lens position coordinates
17	TLY	

After reading the input tape, the basic time step loop of the Pointing Control program is entered. The first calculations made in this loop solve for the theoretical point-ahead angle. Succeeding calculations are performed as indicated in Figure 6-1.

The basic time step loop is recycled through until the elapsed time in this loop equals 500T (the loop has been cycled through 500 times) or mission time has elapsed. Time is advanced by T seconds (presently 0.01) each pass through the basic time step loop. If the mission time duration for which simulation is desired (QUIT) is greater than 500T seconds, simulation is broken up into K, 500 step increments, where K is the largest integer such that K*500T is equal to or less than QUIT. After having read records off tape and cycled through the basic time step loop in K, 500 step increments (and K*500T is not equal to QUIT) the "read loop" of the program is finally entered for the last time and n records read into core (where n = QUIT-K*500T)/T). The basic time step loop is re-entered and cycled through n times to complete simulation of the mission for the requested time duration.

Execution of the program in this fashion is necessitated because of core requirements. In general, it is not possible to read and store all the input data at one time. Consequently, the data is read, stored, and processed in 500 record blocks until the number of records remaining on tape is less than 500, at which time the remaining records are read and processed.

Selected variables are printed at a frequency determined by program word PRCNTL. The output, as presently programmed is discussed in Paragraph 6.3.3. The selected quantities are printed once for every PRCNTL passes through the basic time step loop.

6.3 PROGRAM USAGE

The following paragraphs summarize some of the foregoing statements related to program usage and present the information necessary to make use of the Pointing Control program.

6.3.1 Hardware and Software Requirements

The Pointing Control program runs on the IBM 7094 computer described in Section 4, on which the LASIM program runs. Access to one input tape is requried with the Pointing Control program. To provide flexibility, this tape is referred to symbolically in the program as PATAPE, which necessitates eliminating buffer areas for all but the tape unit assigned to PATAPE. This is done using $FILE cards as indicated in the deck setup in Paragraph 6.3.4. Presently, the input tape designation, PATAPE, is set equal to logical unit 10; which, from Table 4-1, has the physical unit designation A6.

The Pointing Control program runs under the IBSYS operating system using the IBJOB processor. Input and output is under control of IOCS. The following system subroutines and library functions are required by the Pointing Control program.

DSIN	DATAN2	DSQRT
DCOS	MOD	

These are all standard FORTRAN IV subroutines.

6.3.2 Program Input

As discussed extensively throughout Section 6, an input tape is required with the Pointing Control program. This tape is generated by the LASIM program. Data is stored on the tape in 17 word records, the sequence of which is indicated in Table 6-1. As presently programmed, the input is mounted on physical unit A6 and is referred to as logical unit 10.

Two program variables must be initialized to the corresponding values used in the LASIM simulation run from which the input tape was generated. These variables are:

- QUIT: An integer word which is equal to the mission time in seconds over which the simulation will be made.
- T: A double precision word which equals the time in seconds of the basic time step increment.

Data cards containing the values of these variables must be included at the end of the program as described below and in Paragraph 6.3.4.

Table 6-2 lists the program variables which are read off data cards in the indicated order and may be varied by the user by changing the value on the data card. Each data card will contain the value for only one variable. The format to be used and the card image is fixed for each variable as indicated.

6.3.3 Program Output

The variables which are so indicated in the Pointing Control Program Dictionary are printed at the frequency specified by program word PRCNTL. All the variables indicated to be printed are printed once for every PRCNTL passes through the basic time loop. Plot output is not provided from the Pointing Control program.

TABLE 6-2. POINTING CONTROL PROGRAM INPUT DATA CARDS

Card Number	Program Variable Name	Format	Description	
1	QUIT	I5	Mission time over which simulation is to be made in seconds.	10 (columns 1–5); blanks (columns 6–80) ← Card Column
2	T	D12.5	Basic time step for simulation in seconds.	.01D0 (columns 1–12); blanks (columns 13–80) ← Card Column
3	PRCNTL	I5	Print frequency, loops per print.	10 (columns 1–5); blanks (columns 6–80) ← Card Column
4	OMEGA	D12.5	Risely prism servo break frequency in radians per second.	72.1D0 (columns 1–12); blanks (columns 13–) ← Card Column
5	ZETA	D12.5	Risely prism servo damping ratio.	1.1D0 (columns 1–12); blanks (columns 13–80) ← Card Column

6.3.4 Deck Setup and Computation Time

Figure 6-2 illustrates the deck setup for the Pointing Control program. As indicated in the figure, $FILE cards are included to inhibit buffer formation for tape units which are not used. These cards follow the normal control cards required by the IBSYS system. It is important to note that a $FILE card should not be used for the tape unit assigned to PATAPE.

Either a $IBFTC card or $IBLDR card must precede the program deck depending upon whether source or object decks respectively are used.

A $DATA card must follow the program deck. Following this, a data card must be included for each of the program variables indicated in Table 6-2 and discussed in Paragraph 6.3.2. The format for the data on the card is fixed and illustrated in Table 6-2.

The last card of the completed program deck must be an end of file card (7 and 8 punches in Column 1).

A sample instruction card is indicated in Figure 6-3. For the sample shown, the input tape is mounted on physical unit A6.

For the nominal basic time step of 0.01 second, the Pointing Control program runs at about a .01:1 ratio of computation time to mission time. This assumes a print frequency of 10 loops per print (PRCNTL = 10). This ratio will increase slightly for longer mission times since the input tape must be read more times at execution time.

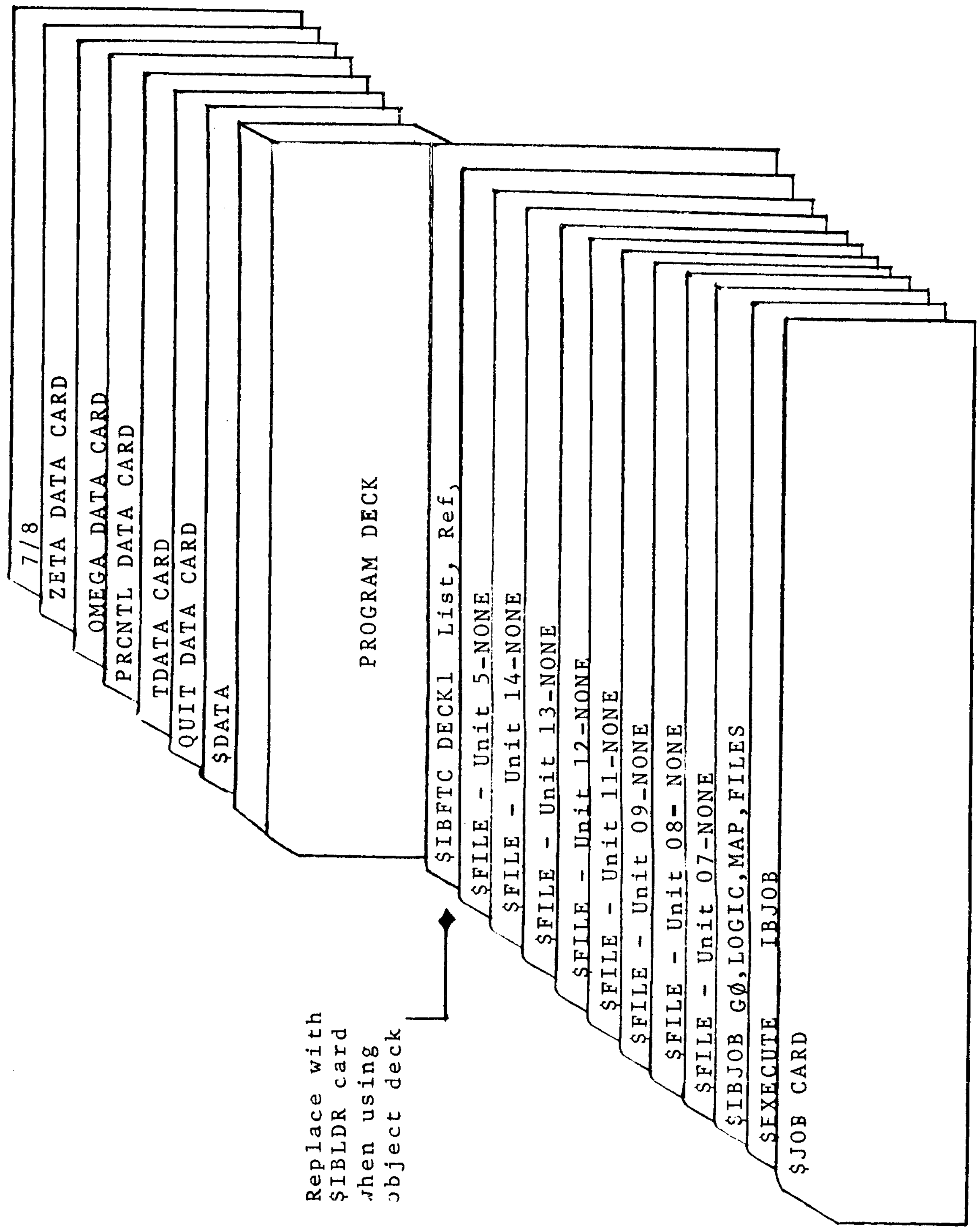


FIGURE 6-2. POINTING CONTROL PROGRAM DECK SETUP

7094-__________ INSTRUCTIONS

NAME: Joe Doe	OP CODE: 2	STACK#
BIN# 217 / LOC BLDG 4200	JOB# 478590	

IF EXCEEDS MAX:

☒ STR ☐ STZ ☐ DMP ☐ RETSY

FAST TAPES: A B C D

☒ IBSYS ☐ SPOOK ☐ OTHER	☒ COMPL /ASSMBL ☐ EXECUTE ☐ PUNCH (BCD BIN)
☒ 4 FTRN ☐ 2 FTRN ☐ APT ☐ PERT	☐ MAP ☐ FAP ☐ SCAT ☐ OTHER

INPUT TAPES			WORK
LOGIC	REEL NO.	DEN	LOGIC
A6		8	

LINES OF OUTPUT (1000'S)

☐ 0-5 ☐ 5-15 ☒ 15-30 ☐ OVER

MAXIMUM TIME:

HOURS 0 MINUTES 5

PROGRAMMER COMMENTS:

NUMBER OF CASES ______

OVER: ______

☐ SEE ON-LINE
☐ SEE TECHNIQUE
☐ MAX EXCEEDED
☐ RETURN TO SYS
☐ LINE MAX

OPERATOR COMMENTS:

OPER INIT: ______
OVER: ______

OUTPUT TAPES ONLY						4020
REEL NO.	LOGIC	DEN.	UNIT	NO OF CPYS	SAVE	TAPE
	B-1	8				

NO FILES	NO FRAMES	COPIES		DENSITY		COPY-FLO		KALVAR
		P	F	5	8	P	F	

MSFC - Form 533 (Rev February 1966)

FIGURE 6-3. POINTING CONTROL PROGRAM USER'S INSTRUCTION CARD

6.4 POINTING CONTROL PROGRAM DICTIONARY

Program Name	Math Symbol	Form	Type	Printed	Nominal Value	Description and Units
AO	c_{op}/d_{op}	V	DP	No	--	Risely prism difference equation coefficient.
A1	c_{1p}/d_{op}	V	DP	No	--	Risely prism difference equation coefficient.
A2	c_{2p}/d_{op}	V	DP	No	--	Risely prism difference equation coefficient.
AARG	—	V	DP	No	--	Intermediate variable equal to Cos (θ'_{PA}).
ANG1	θ'_{PA}	V	DP	No	--	Prism rotation to produce desired point ahead cycle.
ARG	θ_{PA1}/M	V	DP	No	--	Intermediate variable equal to Sin (θ'_{PA})
B	b	VR	DP	No	--	Intermediate vector in calculation of unit vector in S_n direction.
B1	$-d_{1p}/d_{op}$	V	DP	No	--	Risely prism difference equation coefficient.
B2	$-d_{2p}/d_{op}$	V	DP	No	--	Risely prism difference equation coefficient.
BETA	β	V	DP	No	--	Intermediate variable in calculation of difference equation coefficients.
BLOOP	—	V	I	No	--	Counter in program for number of tape records to be used.
C	S_u	VR	DP	No	--	Unit vector in direction of projection of sun vector normal to line-of-sight.
COUNT	—	V	I	No	--	Counter in program equal to total number of records read from tape.

Program Name	Math Symbol	Form	Type	Printed	Nominal Value	Description and Units
CPA	$\text{Cos}(\theta_{PA})$	V	DP	No	--	Trignometric cosine of θ_{PA}.
DELT	$\text{Sin}^2(\theta_R)$	V	DP	No	--	Intermediate variable.
DESX	--	V	DP	No	--	X coordinate of image of desired down-link beam in f/15 plane.
DESY	--	V	DP	No	--	Y coordinate of image of desired down-link beam in f/15 plane.
DIV	609.601/6M	V	DP	No	--	Conversion factor.
DOT	$S_u \cdot P_u$	V	DP	No	--	Dot product of unit vectors S_u and P_u $= \text{Cos}(\theta_R)$.
DOWNX	$P_{ax} \cdot 609.601$	V	DP	No	--	X coordinate of image of actual down-link beam in f/15 plane.
DOWNY	$P_{ay} \cdot 609.601$	V	DP	No	--	Y coordinate of image of actual down-link beam in f/15 plane.
ERRX	ε_x	V	DP	Yes	--	Total pointing error x component in arc seconds.
ERRY	ε_y	V	DP	Yes	--	Total pointing error y component in arc seconds.
F	x_p	VR	DP	No	--	Unit vector defining the x axis of the [P] frame.
G	y_p	VR	DP	No	--	Unit vector in direction of $\dot{L}_S \times L_S$, [P] frame y axis.
GAMMA	γ	V	DP	No	--	Intermediate variable in calculation of difference equation coefficients.
LAMDA	λ	V	DP	No	--	Intermediate variable in calculation of difference equation coefficients.

Pointing Control Program Dictionary (Continued)

Program Name	Math Symbol	Form	Type	Printed	Nominal Value	Description and Units
LCOUNT	—	V	I	No	--	Number of read loops in which 500 records will be read.
LMAG	$\|L_S\|$	V	DP	No	--	Magnitude of line-of-sight vector.
LMX	L_X	V	DP	No	--	X component of line-of-sight vector.
LMY	L_Y	V	DP	No	--	Y component of line-of-sight vector.
LMZ	L_Z	V	DP	No	--	Z component of line-of-sight vector.
LOOP	—	V	I	No	--	Counter in program set equal to BLOOP.
LSITE	Z_p	VR	DP	No	--	Unit vector in direction of L_S [P] frame Z axis.
OCOM1	θ_{1C}	V	DP	Yes	--	Risely prism #1 servo command angle, radians.
OCOM2	θ_{2C}	V	DP	Yes	--	Risely prism #2 servo command angle, radians.
OMEGA	ω	V	DP	Yes	72.1 r/sec	Risely prism servo natural break frequency.
OP1	θ_{P1}	V	DP	Yes	--	Risely prism #1 output, prism #1 rotation, radians.
OP2	θ_{P2}	V	DP	Yes	--	Risely prism #2 output, prism #2 rotation, radians.
OPNTA	θ_{PAa}	V	DP	Yes	--	Actual point ahead angle in prism units.

Pointing Control Program Dictionary (Continued)

Program Name	Math Symbol	Form	Type	Printed	Nominal Value	Description and Units
OTAN	$Tan(\theta_{PAa})$	V	DP	No	--	Intermediate variable.
P	P_1	VR	DP	No	--	Unit vector in desired pointing vector direction in inertial coordinates.
PATAPE	--	V	I	Yes	10	Logical unit designation for input tape.
PRCNTL	--	V	I	Yes	10	Printout frequency, loops per print.
PTEL	$P_{I[T]}$	VR	DP	No	--	P_1 in telescope frame coordinates.
QUIT	--	V	I	Yes	--	Mission time for which simulation is desired in seconds.
SNI	S	VR	DP	No	--	Sun vector in inertial coordinates.
SNO	$S'_{[T]}$	VR	DP	No	-.707, .707,0	Initial sun vector in telescope coordinates.
SNTEL	$S_{[T]}$	VR	DP	No	--	Sun vector in rotating telescope frame coordinates.
SPA	$Sin(\theta_{PA})$	V	DP	No	--	Trignometric Sine of θ_{PA}.
T	--	V	DP	Yes	.01	Basic time step in seconds.
TEMP	$L'_S \times L_S$	VR	DP	No	--	Intermediate variable.
TEMP1	p', $L'_S \times L$	VR	DP	No	--	Relative velocity vector normal to line-of-sight.
THERAC	θ_{Ra}	V	DP	Yes	--	Actual roll reference angle.
THETPA	θ_{PA}	V	DP	Yes	--	Desired point ahead angle.
THETR1	θ_R	V	DP	Yes	--	Desired roll reference angle.

Pointing Control Program Dictionary (Continued)

Program Name	Math Symbol	Form	Type	Printed	Nominal Value	Description and Units
TMAG	$\lvert L'_S \times L_S \rvert$	V	DP	No	--	Magnitude of vector TEMP.
TMAG1	$\lvert \dot{p} \rvert$	V	DP	No	--	Magnitude of vector TEMP1.
TSQ	--	V	DP	No	--	Intermediate variable = T squared.
XTIME	--	V	I	Yes	--	Elapsed time in the simulation; # loops x T.
ZETA	ζ	V	DP	Yes	1.0	Risely prism servo damping.

LEGEND: Form

A = array, matrix
VR = vector array
C = constant
V = variable

Type

I = integer
DP = double precision
R = real

Pointing Control Program Dictionary (Continued)